

**UNIVERSIDAD AUTÓNOMA DE MADRID**

**ESCUELA POLITÉCNICA SUPERIOR**



**TRABAJO FIN DE MÁSTER**

# **SIMULACIÓN Y VERIFICACIÓN DE UN CONVERTIDOR INVERSOR MEDIANTE TÉCNICAS HIL**

**Programa Oficial de Postgrado en Ingeniería  
Informática y de Telecomunicación**

**ALEJANDRO GARCÍA TALÓN**

**Septiembre 2014**



# **SIMULACIÓN Y VERIFICACIÓN DE UN CONVERTIDOR INVERSOR MEDIANTE TÉCNICAS HIL**

ESTUDIANTE:

**ALEJANDRO GARCÍA TALÓN**

TUTOR:

**DR. ALBERTO SÁNCHEZ GONZÁLEZ**





## **Agradecimientos**

A Ángel, mi tutor desde el inicio del máster y amigo durante muchos años. A Alberto, por ser mi tutor durante este trabajo, mi consejero y por la inmensa ayuda prestada (incluidos los viernes por la tarde). A mi familia y a Noe por todo su apoyo y cariño.



## Resumen

El uso de reguladores digitales para controlar convertidores de potencia en lugar de sus equivalentes analógicos es cada vez mayor, debido a las innegables ventajas que presenta el uso de circuitos digitales. Aun así presentan ciertos inconvenientes, como por ejemplo la dificultad de depurar el sistema de control, ya que nos encontramos con que la planta es de naturaleza analógica y el regulador es un dispositivo digital. Por tanto, la depuración de sistemas mixtos que emplean señales analógicas y digitales no es trivial.

Un método para realizar la depuración de sistemas mixtos es la emulación HIL (*Hardware-In-the-Loop*). La emulación HIL es una técnica utilizada para el desarrollo y verificación de sistemas embebidos en tiempo real, y consiste en modelar la planta o circuito de potencia a controlar e implementar dicho modelo en un dispositivo digital programable junto a los elementos de control. De esta forma se evita tener que realizar verificaciones sobre la planta real que puedan provocar daños en los prototipos, equipos o las personas, con una velocidad y precisión que no pueden alcanzarse con otros medios de simulación convencionales.

En este trabajo se describe un sistema de verificación, mediante técnicas HIL y utilizando el lenguaje VHDL (*Very High Speed Integrated Circuit HDL*), del control de un circuito electrónico de potencia, en concreto, un convertidor inversor DC-AC monofásico con topología de puente en H o *full-bridge*.

El dispositivo digital empleado como plataforma para el sistema HIL en este trabajo es una FPGA (*Field Programmable Gate Array*). Para implementar el modelo HIL de planta se utilizan señales en formato de coma fija o notación QX.Y. Por otra parte, también se presenta un modelo en coma flotante, de alta precisión pero con menor velocidad de ejecución, ya que no es emulable. Por tanto, este trabajo compara ambos modelos, para así comprobar la validez y utilidad del modelo HIL, analizando aspectos como el diseño del modelo de la planta, su velocidad de simulación y emulación, precisión, etc.

## Palabras Clave

Sistema de control, control digital, FPGA, simulación, emulación, convertidor inversor, *hardware in-the-loop*, QX.Y, coma fija.

## **Abstract**

The use of digital controllers rather than their analogue equivalents for controlling power electronic converters is increasing due to the advantages of using digital devices. Even so there are some drawbacks, for instance, the difficulty of debugging the control system, because the analogue nature of the plant and the controller being digital device. Therefore the debugging of mixed analogue/digital signals systems is not a trivial task.

One method for debugging mixed signal systems is the HIL (Hardware-In-the-Loop) emulation technique. Hardware in-the-loop simulation is a technique used in the development and verification of real-time embedded systems that consists of designing and modeling digitally the power electronic circuit or plant, and implementing that model along with the control elements in a digital programmable device. In that way, the need of a physical prototype of the plant under control can be avoided for the testing, and with that, the consequent risk of damage in the prototype, instruments or humans.

This project presents a verification system for the control of a power electronics circuit, namely, a single phase full-bridge DC-AC power inverter, using HIL verification techniques

The device used as a platform for the HIL system is an FPGA (Field Programmable Gate Array). For implementing the plant model fixed-point notation or QX.Y signals are used. Besides, a high accuracy floating point model is also presented. That system is slower because it cannot be emulated. The two systems are compared in this work to prove the performance and validity of the HIL system, analyzing aspects such as the design of the plant model, the simulation and emulation speeds, accuracy, etc.

## **Key words**

Control system, digital control, FPGA, simulation, emulation, power inverter, hardware in-the-loop, QX.Y, fixed point.

## Índice General

<b>Agradecimientos .....</b>	<b>III</b>
<b>Índice de Figuras .....</b>	<b>VIII</b>
<b>Índice de Tablas .....</b>	<b>XI</b>
<b>1. Introducción.....</b>	<b>1</b>
1.1 Motivación del trabajo .....	1
1.2 Objetivos y alcance del trabajo fin de master .....	2
1.3 Organización de la memoria .....	3
<b>2. Estado de La Técnica .....</b>	<b>5</b>
<b>3. Control del Convertidor Inversor .....</b>	<b>9</b>
3.1. Topología del Inversor.....	9
3.2. Función de transferencia de la planta .....	11
3.3. Función de transferencia del regulador .....	15
<b>4. Modelado de la planta .....</b>	<b>19</b>
4.1 Ecuaciones en diferencias .....	19
4.2 Implementación del modelo de la planta con señales de tipo <i>Real</i> .....	22
4.3 Implementación del modelo de la planta con señales en coma fija .....	23
4.3.1 Escalado de señales en coma fija .....	24
4.4 Implementación del regulador con señales en coma fija.....	30

<b>5. Resultados .....</b>	<b>33</b>
5.1 Descripción del sistema completo. ....	34
5.2 Precisión del sistema de emulación. ....	35
5.3 Velocidad de simulación y emulación. ....	42
5.4 Comparativa .....	43
<b>6. Conclusiones y trabajo futuro.....</b>	<b>45</b>
6.1 Conclusiones .....	45
6.2 Trabajo futuro .....	46

## Índice de figuras

1.1. Simulación de un sistema de naturaleza mixta analógico/digital .....	1
2.1. Simulación mixta con señales analógicas/digitales .....	6
2.2. Esquema de simulación HIL para verificación de reguladores digitales.....	7
3.1. Topología del convertidor inversor <i>full-bridge</i> .....	10
3.2. Ramas de corriente en un convertidor inversor. ....	12
3.3. Captura de pantalla de la herramienta <i>SisoTool</i> empleada en el cálculo de la función del regulador.....	16
3.4 Respuesta al escalón de la planta discretizada y del sistema en lazo cerrado.....	16
4.1. Tensiones de salida y en la bobina y corriente en la bobina en el convertidor inversor. ....	20
4.2. Circulación de la corriente durante los estados intermedios de conmutación.....	21
4.3. Diagrama del modelo de la planta implementado en VHDL con los formatos de las señales y escalas aplicadas .....	29
4.4. Diagrama del regulador implementado en VHDL y formatos de las señales .....	32
5.1. Diagrama de bloques del sistema de emulación completo. ....	35
5.2. Formas de onda para la tensión de salida, corriente en la bobina y corriente en la resistencia usando señales de tipo <i>real</i> . ....	36
5.3. Formas de onda para la tensión, corriente en la bobina y corriente en la resistencia usando un regulador sintetizable y señales de tipo <i>real</i> para el modelo de la planta.....	37
5.4. Formas de onda para la tensión de salida, corriente en la bobina y en la resistencia usando señales en formato QX.Y .....	37
5.5. Formas de onda del modelo en lazo abierto con ciclo de trabajo 0,5 usando señales <i>real</i> (a) y en formato QX.Y (b).....	39
5.6. Formas de onda del modelo en lazo abierto con ciclo de trabajo 0,1 usando señales <i>real</i> (a) y en formato QX.Y (b).....	39
5.7. Formas de onda del modelo en lazo abierto con ciclo de trabajo 0,9 usando señales <i>real</i> (a) y en formato QX.Y (b).....	40
5.8. Tensión y corriente de salida en el inversor frente a distintos valores de tensión de referencia .....	41
5.9. Saturación del inversor al aplicar un valor excesivo de tensión de referencia .....	41





## Índice de tablas

3.1. Valores escogidos para distintos parámetros de la planta .....	15
4.1. Estado de los transistores en cada rama de funcionamiento del inversor .....	21
4.2. Formato, escala, resolución y rango de las señales empleadas en el modelo de planta. ....	27
5.1. Parámetros de diseño escogidos del entorno de simulación HIL.....	36
5.2. Valores de tensión de salida para el modelo en lazo abierto en los sistemas con señales de tipo real y con señales en formato QX.Y .....	38
5.3. Valores de corriente en la bobina para el modelo en lazo abierto en los sistemas con señales de tipo real y con señales en formato QX.Y .....	38
5.4. Valores de tensión de salida para el modelo en lazo cerrado en los sistemas con señales de tipo real y con señales en formato QX.Y. ....	41
5.5. Valores de tensión de salida para el modelo en lazo cerrado en los sistemas con señales de tipo real y con señales en formato QX.Y. ....	41
5.6. Utilización de recursos de la FPGA para el modelo de planta y para el sistema HIL completo .....	42



# 1

## Introducción

### 1.1 Motivación del trabajo

El uso de dispositivos digitales para realizar el control de circuitos electrónicos de potencia está cada vez más extendido debido a las grandes ventajas que aportan respecto a sistemas analógicos similares (flexibilidad, reprogramabilidad, velocidad, etc.) [1]. Pero aun así, su uso es menor en aplicaciones comerciales reales respecto a los analógicos, debido a factores tales como su mayor dificultad de diseño, y por ello siguen siendo hoy en día objeto de investigación.

Una de las principales dificultades a la hora del diseño de un sistema de control digital es la simulación para la depurar su correcto desempeño, lo cual es muy importante cuando se trabaja con circuitos de alta potencia, pues cualquier fallo en el controlador puede causar daño en el sistema, los instrumentos o las personas. Esta dificultad en la depuración del controlador es debida a la naturaleza mixta de las señales en el sistema: la planta está compuesta de elementos pasivos y semiconductores y por tanto es de naturaleza analógica y el controlador es digital (figura 1.1). Como se explicará adelante existen numerosas herramientas que permiten realizar fácilmente simulaciones de sistemas completamente analógicos, pero las simulaciones mixtas con señales analógicas y digitales no son tan sencillas.



Figura 1.1. Simulación de un sistema de naturaleza mixta analógico/digital.

La técnica de depuración HIL consiste en implementar un modelo de la planta o circuito a regular junto con el controlador, y realizar pruebas al sistema completo. En el caso de reguladores descritos con un lenguaje de descripción de *hardware*, un modelo en la planta también realizado en el mismo lenguaje permite usar un único simulador, habiendo además

soluciones de bajo coste. Además, un modelo digital de la planta permite ser emulado en *hardware* real, es decir, ser probado usando un dispositivo como una FPGA o un ASIC (*Application-Specific Integrated Circuit*). Aprovechando la naturaleza concurrente de estos dispositivos, la depuración del funcionamiento del controlador junto a la planta puede realizarse a gran velocidad, incluso en tiempo real, aportando una gran flexibilidad en el diseño y reduciendo los costes y el riesgo de daños.

En [2,3,4] se ha demostrado que es posible la emulación de un regulador digital para controlar un convertidor elevador. En este trabajo fin de Master se va a comprobar si estas técnicas también son viables para un convertidor inversor. Los inversores son ampliamente utilizados en aplicaciones industriales como en plantas solares para transformar la corriente continua generada en paneles fotovoltaicos o de manera creciente en el campo de la automoción en vehículos con motores eléctricos.

## **1.2 Objetivos y alcance del trabajo fin de master**

En este trabajo fin de master se propone la realización de un sistema de depuración para un circuito de potencia mediante técnicas HIL. La planta es en este caso un inversor DC/AC, un tipo de convertidor que transforma una tensión continua de entrada en una tensión sinusoidal de una determinada frecuencia a su salida.

El objetivo principal de este trabajo es diseñar e implementar digitalmente un modelo del inversor y de un sistema de control utilizando lenguaje VHDL y demostrar la validez del método de emulación HIL en circuitos electrónicos de potencia.

Para ello se van a emplear dos aritméticas: coma flotante de doble precisión (utilizando señales de tipo *real*) no sintetizable pero de alta resolución, y coma fija sintetizable (formato QX.Y) la cual va a permitir la emulación del sistema, alcanzando aceleraciones muy grandes. Se va a realizar un estudio comparativo exhaustivo de ambas aritméticas en términos de precisión, tiempo de simulación y emulación, y recursos ocupados en la FPGA en la emulación.

Para comprobar la validez del sistema se incluye el diseño de un regulador sencillo, desde el análisis de la planta hasta la extracción de las funciones de transferencia de la planta y regulador elegido, además de los elementos necesarios para el control del inversor, como son los módulos PWM e interconexiones.

### 1.3 Organización de la memoria

Este trabajo está estructurado de la siguiente manera:

- Tras la introducción se presenta el segundo capítulo donde se describe estado actual de la técnica respecto a la simulación y verificación de controladores digitales para circuitos electrónicos de potencia.
- En el capítulo tercero se trata la técnica empleada para el control del convertidor inversor, describiendo éste en detalle y la obtención de la función de transferencia de la planta necesaria para calcular la del controlador.
- El cuarto capítulo se describe cómo se implementa digitalmente el modelo de la planta así como los métodos empleados para optimizar dicha implementación.
- El quinto capítulo aborda los resultados obtenidos del modelo HIL implementado, así como su validez en términos de resolución y velocidad.
- Finalmente se presentan en el sexto capítulo las conclusiones extraídas de este trabajo y las posibles líneas futuras de investigación relacionadas.



# 2

## Estado de La Técnica

En el proceso de diseñar e implementar un sistema de control digital para un circuito de potencia, ya sea un convertidor inversor o cualquier otro circuito de potencia en general, se hace imprescindible tener un medio de depuración para simular y verificar el sistema antes de comprobar su funcionamiento real. Esto es debido a que se manejan potencias y corrientes elevadas que pueden causar daños si el control no ha sido depurado, no solo en los componentes o hardware empleado o en los sistemas de medición, sino incluso en las personas.

Además la verificación y simulación previa a la construcción de sistemas reales aporta un gran grado de flexibilidad, puesto que se pueden realizar comprobaciones y pruebas que de otro modo dañarían el equipo real, como por ejemplo estudios de estabilidad, variaciones frente a cargas, limitaciones de ciclos de trabajo, diseño de medidas de protección, etc. Por otro lado se reducen los costes de diseño puesto que estas verificaciones se realizan sin riesgo alguno para el equipo en el que se prueban y se pueden realizar con dispositivos *hardware* de coste relativamente bajo.

Para poder verificar el funcionamiento del sistema completo es necesario comprobar el funcionamiento no solo del regulador de manera aislada, sino de su comportamiento frente al circuito de potencia o planta que vaya a regular.

A la hora de diseñar un regulador generalmente se utilizará una herramienta como *Matlab* y el paquete *Sisotool* [5] con la que comprobarán parámetros como la estabilidad, dinámica, ancho de banda, etc. y se obtendrá una función de transferencia del mismo. Con dicha función de transferencia se realizará su implementación en un procesador, un DSP (Digital Signal Processor) o una FPGA o ASIC mediante un lenguaje de descripción *hardware* o HDL (*Hardware Description Language*) como VHDL o *Verilog*.

Como se ha comentado anteriormente, es necesario comprobar el funcionamiento del regulador, no solo del controlador de manera aislada, sino junto con la planta y todos los

componentes necesarios (ADCs, generadores de señal, comparadores, etc.). Esto es recomendable ya que las herramientas de diseño de reguladores permiten elegir un regulador matemáticamente óptimo, pero la implementación del mismo en *software* o *hardware* puede producir errores de transcripción de la función de transferencia, reducción de la resolución numérica del regulador, o incluso detalles de arquitectura como puede ser segmentación en el caso del desarrollo del regulador en *hardware*. Por ello, la depuración HIL permite comprobar si el regulador definitivo funcionará correctamente cuando se pruebe con una planta real.

En este apartado se describen las distintas posibilidades que existen para modelar la etapa de potencia y realizar simulaciones junto con a un regulador digital.

Una primera opción es utilizar un software de simulación mixto analógico-digital [6] (figura 2.1). En este tipo de software la planta se modela mediante un esquemático de manera sencilla con el método de “arrastrar y soltar” componentes. Estos componentes generalmente están internamente descritos en lenguaje VHDL-AMS (*Hardware Description Language – Analog and Mixed Signal*), una extensión del lenguaje HDL diseñado para modelar el comportamiento de sistemas de señales analógicas y mixtas. Los componentes digitales como el regulador y otros descritos en VHDL se añaden como un componente más en el sistema a simular.

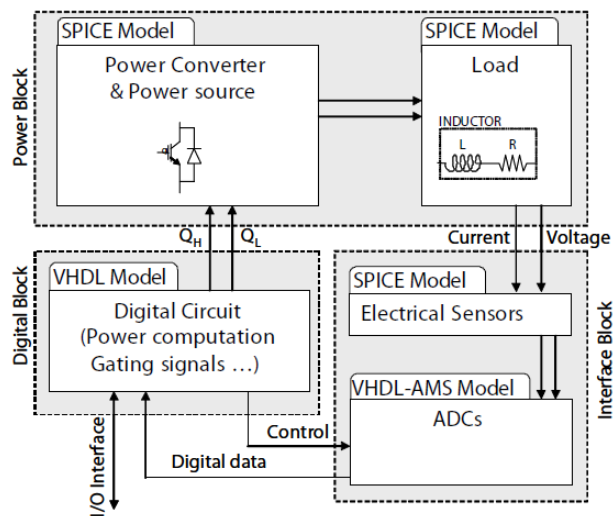


Figura 2.1. Simulación mixta con señales analógicas/digitales

Una de las grandes ventajas de este método de verificación conjunta es que es muy sencillo para el diseñador, pero como inconvenientes presenta que las simulaciones son muy lentas, comparadas con otras alternativas que se analizan más adelante, y además existen pocas herramientas comerciales (a la fecha de creación del presente trabajo fin de Master, Mayo de 2014), siendo la mayoría de ellas de pago. Ejemplos son *Questa Advanced Simulator* [7], de *Menthor Graphics* o la herramienta *SystemVision* [8].



Una segunda alternativa es el uso de dos simuladores diferentes [9], uno para la parte del regulador digital (*ModelSim*) y otro para la parte analógica (PSIM). Esto presenta el inconveniente de que es el propio diseñador el que tiene programar interfaces para la comunicación entre ambos entornos, lo que complica enormemente el proceso de diseño.

La tercera alternativa principal es implementar un modelo de la planta junto con el regulador mediante un lenguaje de descripción *hardware*, como VHDL (figura 2.2). Esto permite realizar simulaciones del modelo HDL con herramientas como *ModelSim*, pero también permiten realizar una emulación del sistema completo en un dispositivo programable, generalmente una FPGA [10, 11, 12], de manera que el regulador es el mismo que se usará en la versión final y la planta es un modelo emulado en un dispositivo digital, lo cual permite una velocidad mucho mayor que la simulación. Este método es lo que se conoce como *hardware in-the-loop* o HIL por sus siglas en inglés y una de sus principales ventajas es la velocidad que se consigue comparada con el método anteriormente descrito, permitiendo emulaciones en tiempo real.

La técnica HIL puede ser utilizada en la verificación en gran variedad de aplicaciones. En [13 y 14] se describe un entorno de emulación con aplicación en cocinas de inducción domésticas de múltiples salidas, utilizando un regulador digital ejecutado en un procesador embebido dentro de una FPGA. En [15] se presenta un método de simulación y emulación mediante técnica HIL de motores de inducción. En [16] se describe un entorno HIL para la verificación de sistemas eólicos de generación de energía.

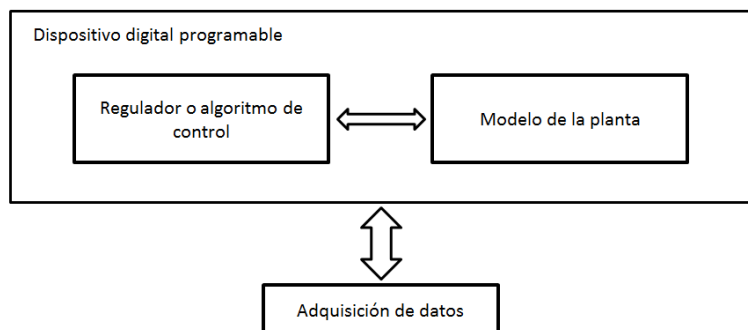


Figura 2.2. Esquema de simulación HIL para verificación de reguladores digitales

Para desarrollar el modelo de la planta es necesario implementar sus ecuaciones características de la planta, siendo factores críticos y no triviales la elección de una aritmética correcta puesto que ello influye en la resolución del sistema, precisión, tiempo de diseño y ejecución y en los recursos *software* o *hardware* necesarios. En el caso de usar un lenguaje HDL para implementar el modelo de la planta, existen varias alternativas que se describen a continuación:

- Usando el tipo de señal real, un tipo numérico con representación en coma flotante de doble precisión, siguiendo el estándar IEEE 754, que permite la simulación y visualización de señales y valores mediante programas como puede ser *ModelSim* de

*Mentor Graphics*. Se puede describir la planta mediante ecuaciones en diferencias de manera rápida y sencilla con este tipo de señal, pero las simulaciones son lentas. Sin embargo, este tipo de datos no es sintetizable y no permite la emulación del sistema.

- Usando el tipo de señal float usando el paquete *float\_pkg* de la librería *VHDL-2008 Support Library*. Es también un tipo numérico con representación en coma flotante del estándar IEEE 754 al igual que el tipo real, pero con la diferencia de que es sintetizable y permite la verificación mediante una emulación del sistema completa. Es relativamente sencillo para el diseñador implementar la planta usando este tipo de señal pero tiene el inconveniente de que existen pocos simuladores y sintetizadores que la soportan. Además, la gran cantidad de recursos *hardware* que necesita hace que no siempre sea útil utilizar este tipo de datos [2].
- Utilizar señales en formato de coma fija. Requiere un esfuerzo de diseño mucho mayor que las opciones anteriores, puesto que el diseñador debe tener en cuenta el formato y ancho de cada una de las señales y del tamaño necesario de las mismas tras cada operación aritmética. Por otro lado es la alternativa que ofrece mejores resultados en cuanto a velocidad de simulación y recursos *hardware* necesarios para implementarla.
- Usando señales en formato de coma fija usando la biblioteca *sfixed* de la librería *VHDL-2008 Support Library*, similar a la anterior pero facilitando las operaciones aritméticas entre señales, permitiendo conversiones de formatos de manera sencilla. Aun así el control sobre el tamaño de cada señal y de cada registro es labor del diseñador al igual que en apartado anterior.

La elección de una de las alternativas descritas anteriormente dependerá del compromiso que se quiera alcanzar entre diversos factores tales como el esfuerzo de diseño, tiempo requerido de simulación o emulación, resolución del modelo y coste en *hardware*.

Como se ha comentado en el capítulo 1 uno de los objetivos de este trabajo fin de Máster es implementar y demostrar el controlador de un convertidor inversor y verificarlo mediante técnicas HIL.

En el caso de un convertidor de potencia, la velocidad de simulación es un factor crucial ya que la frecuencia de las señales de salida que se van a comprobar son del orden de hercios, pero por otro lado se necesitan también realizar suficientes cálculos dentro de cada periodo de conmutación para obtener una resolución adecuada siendo la frecuencia de conmutación del orden de kilohercios y el reloj del sistema del orden de megahercios. Por tanto, es necesario que el modelo de la planta tenga un tiempo de integración muy pequeño, lo que conllevará realizar millones de ciclos de integración para depurar un regulador, haciendo que la depuración sea costosa. Por ello, en este trabajo se realiza una comparación en términos de velocidad de simulación de un modelo que utiliza señales de tipo real y un modelo que usa señales en formato de coma fija, con el fin de demostrar la validez de un sistema HIL para la verificación del controlador digital de un convertidor inversor.

# 3

## Control del Convertidor Inversor

Como se ha comentado en el capítulo anterior en este trabajo de fin de Máster se pretende demostrar y desarrollar un sistema de verificación HIL del regulador digital de un convertidor inversor modelado mediante un lenguaje de descripción *hardware*. En este capítulo se estudian las distintas topologías de convertidores inversores y el sistema de regulación desarrollado, tanto en lazo cerrado como en lazo abierto.

### 3.1. Topología del Inversor.

Un convertidor inversor es un dispositivo que permite transformar una corriente continua aplicada a su entrada en una corriente alterna senoidal de una amplitud y frecuencia determinadas. Se llaman inversores porque funcionan en sentido inverso al que típicamente tienen los convertidores de potencia que transforman corrientes alternas en continuas. Un ejemplo de aplicación de los inversores es su uso en paneles solares fotovoltaicos. Los inversores que funcionan de manera autónoma sin necesidad de estar conectados a ninguna red de alterna, o también llamados inversores u onduladores autónomos o autoguiados, son en los que nos centraremos en este trabajo fin de Máster.

Un inversor generalmente consta de una fuente de tensión continua y varios interruptores que se conmutan de manera alternada para obtener semiciclos positivos y negativos. Existen diversas configuraciones o topologías de inversores que se describen a continuación:

- Topología *Push-Pull*. Se trata de un circuito con un transformador con toma intermedia a la cual está conectada el polo positivo de una fuente de corriente continua y dos conmutadores que conectan los polos del devanado primario con el polo negativo de la fuente de tensión continua. Mediante la conmutación alternativa de los dos

interruptores se obtiene a la salida una onda de tensión cuadrada cuya frecuencia depende directamente de la de conmutación.

- Topología de tipo *half-bridge* o semi-puente. Consta de una fuente de alimentación continua simétrica y dos transistores  $Q_1$  y  $Q_2$  que conmutan alternativamente. Cuando  $Q_1$  está abierto y  $Q_2$  cerrado la tensión aplicada en la carga es de  $-V_g/2$ . En el caso inversor la tensión es de  $V_g/2$ . Se añaden además diodos en antiparalelo para permitir la circulación de corriente en el caso de cargas reactivas.
- Finalmente la topología *full bridge* o de puente completo o puente en H es una extensión del anterior con cuatro MOSFET (*Metal-Oxide-Semiconductor Field-Effect Transistor*) tal y como se muestra en la figura 3.1. Para que el circuito funcione adecuadamente se debe asegurar una trayectoria para la corriente de la bobina en cualquier instante de tiempo, lo cual implica que se deben encender dos de los cuatro transistores del circuito. En este caso se conmutan alternativamente los transistores  $Q_1$ - $Q_2$  y  $Q_3$ - $Q_4$  creando la rama 1 y rama 2 respectivamente. En el primer caso ( $Q_1$ - $Q_2$  cerrados) la corriente circula por la rama 1 y la tensión en la red R-L-C es igual a  $V_g$ , y en el caso opuesto ( $Q_3$ - $Q_4$  cerrados) la tensión en la red es  $-V_g$ .

Aplicando un ciclo de trabajo  $D$  sobre el periodo de conmutación en la a los MOSFET de la rama 1 y un ciclo de  $1-D$  sobre los de la rama 2 la carga queda sometida a una tensión alterna cuadrada de amplitud igual a la de la fuente continua y de frecuencia la misma que la de conmutación.

En este trabajo la topología empleada para la planta será la del puente en H completo, aunque el desarrollo y resultados de este trabajo fin de Máster serían similares en el caso de las otras topologías mencionadas.

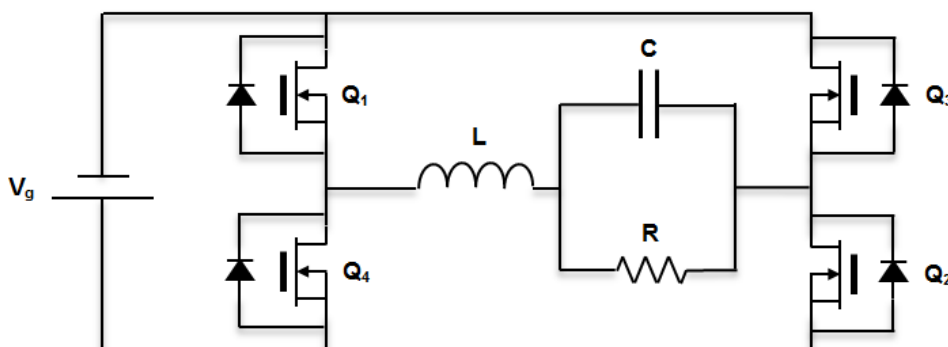


Figura 3.1. Topología del convertidor inversor *full-bridge*

### 3.2. Función de transferencia de la planta

Uno de los requisitos prácticos para un convertidor inversor es mantener constante el valor eficaz de la tensión de salida ante variaciones en la carga o la tensión de entrada y eso puede conseguirse mediante distintos procedimientos: controlando la tensión continua en la entrada, regular la tensión de salida o realizar una regulación interna en el propio inversor utilizando señales PWM (*Pulse Width Modulation*), siendo este último el más eficiente y el que se ha utilizado para el regulador del sistema planteado en este trabajo.

La regulación se hace controlando el ancho de pulso de las señales que gobiernan los interruptores del convertidor en función del voltaje de salida. Se trata por tanto de un sistema de control en lazo cerrado siendo el voltaje de salida la señal realimentada. En un sistema de control el regulador es el componente que realiza la labor de control y regulación sobre el elemento a controlar o planta, en nuestro caso, el convertidor inversor. Por tanto, para obtener un modelo del regulador necesitamos primero conocer la función de transferencia de la planta, es decir, la parte de potencia que se va a regular, tal y como se explica a continuación.

En la planta  $R$  representa el valor de una carga resistiva conectada al inversor y  $L$  y  $C$  la inductancia y capacidad del filtro de salida respectivamente. Partimos del valor de tensión en la bobina:

$$v_L = L \frac{di}{dt}$$

Cuando se encuentran cerrados los interruptores  $Q_1$  y  $Q_2$  y los interruptores  $Q_3$  y  $Q_4$  se encuentran abiertos, es decir, está activa la rama 1 del convertidor (figura 3.2 a), tenemos que la variación de la corriente en la bobina es:

$$\Delta i_1 = (V_g - V_{out}) \frac{dT_s}{L}$$

Siendo  $T_s$  el periodo de la señal de conmutación y  $d$  el valor del ciclo de trabajo de la señal de conmutación. En la ecuación se ha considerado que  $V_g$  es constante y que  $V_{out}$  también es constante durante un periodo de conmutación. Es cierto que  $V_{out}$  es sinusoidal, pero su frecuencia es muy baja (en torno a las decenas de hercios) mientras que la frecuencia de conmutación es alta (en torno a los kilohercios). De forma análoga, cuando se da el caso opuesto y la rama 2 (figura 3.2 b) se encuentra activa tenemos que el incremento de corriente en la bobina durante ese tiempo es:

$$\Delta_{i2} = (-V_g - V_{out}) \frac{(1-d)T_s}{L}$$

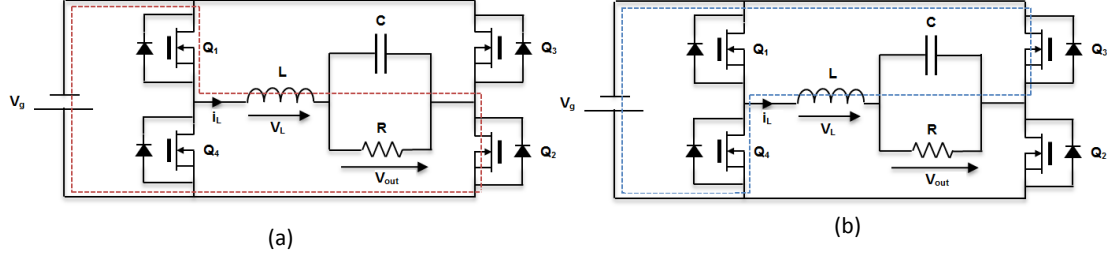


Figura 3.2. Ramas de corriente en un convertidor inversor.

En régimen permanente podemos decir que ambos valores de corriente son iguales,  $\Delta_{i1} = \Delta_{i2}$  y por tanto:

$$(V_g - V_{out}) \frac{dT_s}{L} + (-V_g - V_{out}) \frac{(1-d)T_s}{L} = 0$$

Agrupando coeficientes tenemos que el valor de la tensión de salida en régimen permanente es:

$$V_g D - V_{out} D - V_g + V_g D - V_{out} + V_{out} D = 0$$

$$2V_g D - V_g - V_{out} = 0$$

$$V_{out} = V_g (2D - 1)$$

La ecuación anterior relaciona la tensión de entrada y la tensión de salida nominales teniendo en cuenta el ciclo de trabajo aplicado a los interruptores del inversor. Para extraer la función de transferencia de la planta y comprobar su dinámica es necesario obtener el modelo del circuito en pequeña señal. Los modelos en pequeña señal se utilizan para analizar convertidores conmutados así como otras plantas y permiten crear un modelo aproximado de plantas no lineales. Estos modelos se calculan en torno al punto de equilibrio y son válidos para pequeñas perturbaciones. En la notación de pequeña señal, una variable está determinada como:

$$\langle v \rangle = V + \hat{v}$$

Donde  $V$  es el valor en el punto de equilibrio de la variable,  $\hat{v}$  es una pequeña perturbación aplicada a la variable, y  $\langle v \rangle$  es la componente resultante de sumar las dos magnitudes anteriores.

Aplicando el modelo de pequeña señal a las ecuaciones anteriormente descritas, se puede obtener las siguientes ecuaciones:

$$L \frac{\partial \langle i \rangle}{\partial t} = (2\langle d \rangle - 1)V_g - \langle v_{out} \rangle$$

$$C \frac{\partial \langle v_{out} \rangle}{\partial t} = \langle i \rangle - \frac{\langle v_{out} \rangle}{R}$$

Siendo:

$$\langle i \rangle = I + \hat{i}$$

$$\langle v_{out} \rangle = v_{out} + \hat{v}_{out}$$

$$\langle d \rangle = D + \hat{d}$$

Se ha considerado que  $V_g$  es constante, por lo que no se ha aplicado el modelo de pequeña señal en su caso. Sustituyendo las equivalencias anteriores tenemos que:

$$L \frac{\partial (I + \hat{i})}{\partial t} = (2(D + \hat{d}) - 1)V_g - (v_{out} + \hat{v}_{out})$$

$$C \frac{\partial (v_{out} + \hat{v}_{out})}{\partial t} = (I + \hat{i}) - \frac{v_{out} + \hat{v}_{out}}{R}$$

Para extraer la función de transferencia, sólo se deben tener en cuenta las perturbaciones de las ecuaciones anteriores:

$$L \frac{\partial \hat{i}}{\partial t} = 2\hat{d}V_g - \hat{v}_{out}$$

$$C \frac{\partial \hat{v}_{out}}{\partial t} = \hat{i} - \frac{\hat{v}_{out}}{R}$$

Si despejamos de la segunda ecuación se tiene que:

$$\hat{i} = C \frac{\partial \hat{v}_{out}}{\partial t} + \frac{\hat{v}_{out}}{R} = C \dot{v}_{out} + \frac{v_{out}}{R}$$

Donde  $\dot{v}_{out}$  representa a la derivada de  $v_{out}$ . Sustituyendo la ecuación anterior en la primera:

$$L \left( C \ddot{v}_{out} + \frac{\dot{v}_{out}}{R} \right) = 2\hat{d}V_g - \hat{v}_{out}$$

$$L C \ddot{v}_{out} + L \frac{\dot{v}_{out}}{R} - \hat{v}_{out} = 2\hat{d}V_g$$

Donde  $v_{out}''$  representa a la derivada doble de  $v_{out}$ . Aplicando Laplace, obtenemos la función de transferencia de la planta:

$$\left( LCs^2 + \frac{L}{R}s + 1 \right) V_{out} = 2dV_g$$

$$\frac{v_{out}(s)}{d(s)} = \frac{2V_g}{LCs^2 + \frac{L}{R}s + 1}$$

Podemos realizar un desarrollo análogo teniendo en cuenta las pérdidas producidas por la resistencia de la bobina  $R_L$  y la resistencia de los transistores  $R_{on}$ . Las ecuaciones del modelo en pequeña señal teniendo en cuenta las pérdidas son las siguientes:

$$L \frac{\partial \langle i \rangle}{\partial t} = (2\langle d \rangle - 1)V_g - (2R_{on} + R_L)\langle i \rangle - \langle v_{out} \rangle$$

$$C \frac{\partial \langle v_{out} \rangle}{\partial t} = \langle i \rangle - \frac{\langle v_{out} \rangle}{R}$$

Que sustituyendo al igual que en modelo sin perturbaciones obtenemos:

$$L \frac{\partial (I + \hat{i})}{\partial t} = (2(D + \hat{d}) - 1)V_g - (2R_{on} + R_L)(I + \hat{i}) - (V_{out} + \hat{V}_{out})$$

$$C \frac{\partial (v_{out} + \hat{v}_{out})}{\partial t} = (I + \hat{i}) - \frac{V_{out} + \hat{v}_{out}}{R}$$

De forma similar al paso anterior nos quedamos con las perturbaciones para obtener:

$$L C v_{out}'' + \left( \frac{L}{R} + C(2R_{on} + R_L) \right) v_{out}' + \left( \frac{2R_{on} + R_L}{R} \right) V_{out} = 2V_g D$$

Finalmente obtenemos la siguiente función de transferencia para la planta teniendo en cuenta pérdidas:

$$\frac{v_{out}(s)}{d(s)} = \frac{2V_g}{LCs^2 + \left( C(2R_{on} + R_L) + \frac{L}{R} \right) s + 1 + \frac{2R_{on} + R_L}{R}}$$



### 3.3. Función de transferencia del regulador

Como se ha comentado en el apartado anterior hemos obtenido la función de transferencia de la planta para conocer su comportamiento y poder obtener así la función del regulador. Se ha optado por discretizar la función de transferencia de la planta y así diseñar directamente el regulador de forma digital. El diseño directo de un regulador discreto es más apropiado que el diseño de un regulador analógico y después discretizarlo. En particular, el diseño directo en digital permite evitar el error de aproximación al discretizar la función de transferencia del regulador previamente analógico, y por otra parte el diseño directo considera de forma inherente los efectos del bloqueador y del muestreador.

El diseño del regulador digital se realizado mediante la utilización de la herramienta *Matlab* y el *toolbox SisoTools*, un complemento gráfico interactivo para el diseño de compensadores y funciones de transferencia en sistemas realimentados.

El primer paso es introducir los parámetros del sistema a controlar, en nuestro caso la función de transferencia de la planta. Inicialmente se definen los valores de los parámetros de la misma. En la tabla 3.1 se muestran los valores de la planta escogidos. Cabe destacar que la función de transferencia que se obtiene es válida para una planta con esos valores concretos. En cualquier caso, en un entorno de validación HIL la planta no suele sufrir modificaciones, siendo el regulador el componente bajo validación sobre el que se realizarán cambios. Por claridad no se han tenido en cuenta las pérdidas en el modelo pero en cualquier caso el desarrollo del regulador y del modelo son similares a los descritos anteriormente.

Con estos valores obtenemos la función de transferencia en dominio continuo de la planta, tras aplicar Laplace con los valores escogidos:

$$G(s) = \frac{90}{9 \cdot 10^{-8} s^2 + 7,5 \cdot 10^{-5} s + 1}$$

Dicha función de transferencia en el dominio del tiempo discreto, aplicando un periodo de discretización de 50  $\mu$ s (frecuencia de conmutación de 20 kHz) corresponde a:

$$G(z) = \frac{0,5467 z + 0,5391}{z^2 - 1,932 z + 0,9592}$$

C(mF)	L(mH)	R <sub>carga</sub> ( $\Omega$ )
0,1	0,9	12

Tabla 3.1. Valores escogidos para distintos parámetros de la planta

Con la planta definida se ajusta de forma gráfica el regulador, añadiendo polos y ceros y modificando sus valores, visualizando los resultados dinámicamente en el gráfico del lugar de

las raíces y diagramas de Bode. El objetivo es que el sistema se mantenga estable, que el regulador elimine el error de la señal a controlar y que tenga una dinámica adecuada. En la figura 3.3 se muestra una captura de pantalla de la herramienta *SisoTool* realizando el cálculo de la función del regulador. La figura 3.4 muestra la respuesta al escalón de la planta discretizada y la respuesta al escalón del sistema en lazo cerrado teniendo en cuenta la función del regulador obtenida.

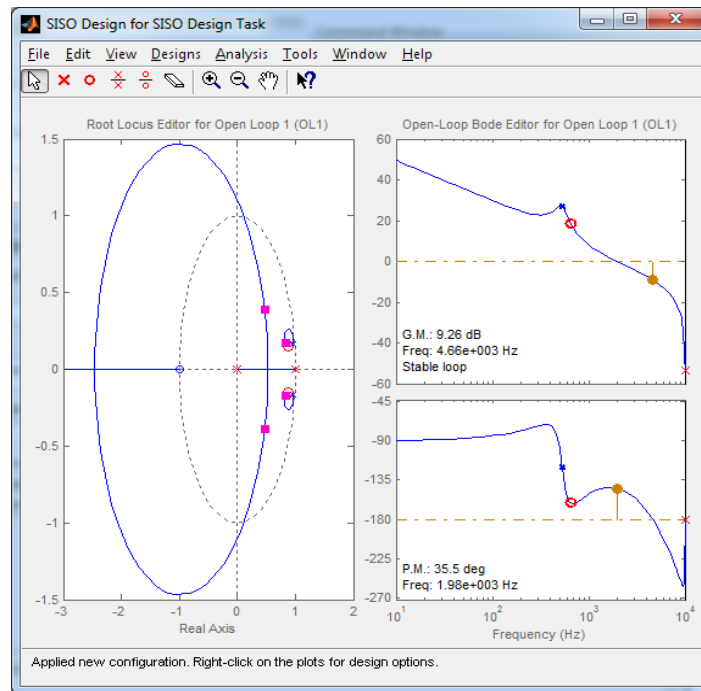


Figura 3.3. Captura de pantalla de la herramienta *SisoTool* empleada en el cálculo de la función del regulador

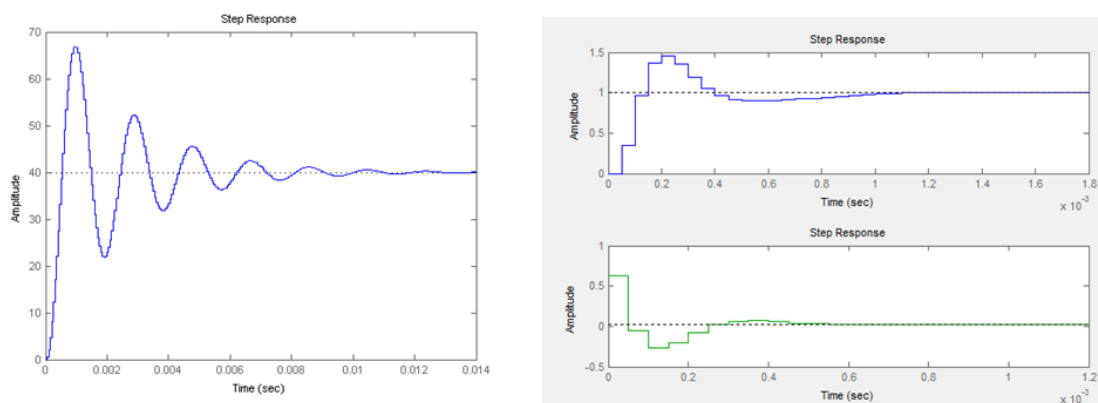


Figura 3.4 Respuesta al escalón de la planta (izquierda) y respuesta de la planta en lazo cerrado (derecha).

La función del regulador de segundo orden obtenida es la siguiente:

$$Reg(z) = \frac{Y(z)}{X(z)} = \frac{A(z^2 + Bz + C)}{z^2 - z} = \frac{A(1 + Bz^{-1} + Cz^{-2})}{1 - z^{-1}}$$

Siendo:

$$A = 0,625$$

$$B = -1,746$$

$$C = 0,7837$$

Se ha elegido un regulador sencillo, priorizando la estabilidad ante el rendimiento, dado que este trabajo fin de Máster se centra en la depuración HIL y no explorar métodos de regulación de un inversor. Además, teniendo en cuenta que el regulador se va a implementar en hardware, los valores de los coeficientes se han ajustado para facilitar su representación en formato binario.



# 4

## Modelado de la planta

En este apartado se muestra cómo modelar el convertidor inversor con topología de puente en H utilizando el formato de señales en coma flotante y en coma fija y las particularidades de dichas señales, para su implementación en un sistema digital de verificación de la etapa de control mediante técnicas HIL.

### 4.1 Ecuaciones en diferencias

La aplicación de un convertidor inversor, como se ha comentado en apartados anteriores, es transformar una señal continua en una señal alterna de amplitud y frecuencia determinadas. El modelo del convertidor debe calcular en cada instante de tiempo la tensión de salida  $v_{out}(t)$  y la corriente de salida que corresponde a la corriente en la bobina  $i_L(t)$ .

La tensión en la bobina es igual a:

$$v_L = L \cdot \frac{di_L}{dt}$$

Donde  $i_L$  es la corriente que circula por la bobina. Transformando la expresión anterior en una ecuación en diferencias obtenemos la siguiente ecuación para la corriente en la bobina:

$$i_L(k) = i_L(k-1) + v_L \cdot \frac{\Delta t}{L}$$

Donde  $i_L(k)$  es la corriente que atraviesa la bobina en el instante de tiempo  $k$  y  $\Delta t$  es el paso de integración en el cálculo de las variables de estado, que en nuestro caso es de 10 ns, y  $v_L$  es la tensión en la bobina.

De forma análoga la corriente que circula por el condensador es:

$$i_C = C \cdot \frac{dv_C}{dt}$$

Siendo  $v_C$  la tensión en el condensador, que corresponde a la tensión de salida  $v_{out}$  y a la tensión en la resistencia. De manera similar a la ecuación anterior, transformando en una ecuación en diferencias, la tensión de salida en el instante de tiempo  $k$  resulta:

$$v_{out}(k) = v_C(k) = v_C(k-1) + i_C \cdot \frac{\Delta t}{C}$$

$i_C$  es la corriente del condensador la cual depende de la carga de salida de manera que  $i_C = i_L - i_R$ . Si la carga solo tiene componente resistiva, la corriente en la resistencia  $R$  se puede calcular como:

$$i_R = \frac{v_{out}}{R}$$

En un convertidor inversor con topología en puente en H, como el que se va a modelar, se emplean cuatro transistores o elementos de conmutación, como se muestra en la figura 4.1, y en su funcionamiento se distinguen dos ramas o estados. El regulador calcula y genera los pulsos necesarios para gobernar los transistores y controlar así la tensión y corriente de salida en el inversor. Las señales de control son del tipo PWM y activan alternativamente las ramas compuestas por los transistores  $Q_1$ - $Q_2$  y  $Q_3$ - $Q_4$  alternativamente. Es crítico que no se activen simultáneamente ambas ramas, pues se produciría un cortocircuito, ni otra combinación como los transistores  $Q_1$  y  $Q_4$  simultáneamente. Sin embargo, no todos los interruptores cambian de estado de conducción a aislamiento y viceversa en el mismo tiempo, sino que habrá ligeras diferencias en esta transición. Para evitar cortocircuitos, se suele añadir un tiempo muerto en las señales de control que gobiernan los transistores para asegurar una conmutación limpia entre una rama y otra.

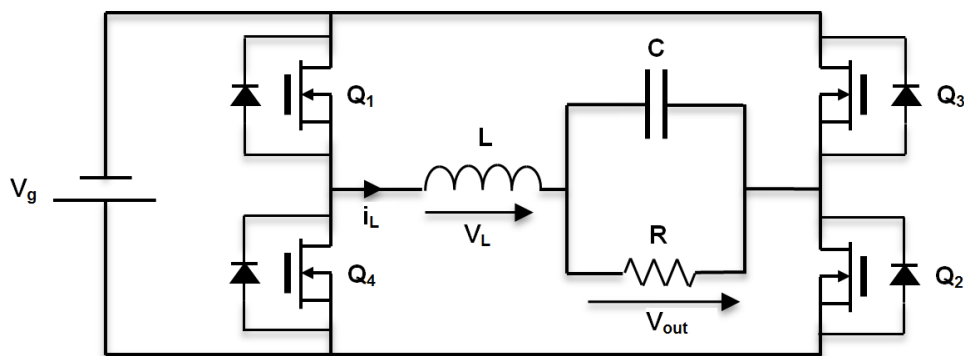


Figura 4.1. Tensiones de salida y en la bobina y corriente en la bobina en el convertidor inversor.

Sean la señal  $M_1$  la señal de control que activa la rama 1 (transistores  $Q_1$  y  $Q_2$ ) y  $M_2$  la señal de control que activa la rama 2 (transistores  $Q_3$  y  $Q_4$ ), según lo anterior podemos tener los estados de control que se muestran en la tabla 4.1.

	$M_1$	$M_2$
<b>tiempo muerto</b>	0	0
<b>rama 1</b>	0	1
<b>rama 2</b>	1	0

Tabla 4.1. Estado de los transistores en cada rama de funcionamiento del inversor

Las ecuaciones que describen el modelo de la planta son diferentes según el estado o rama que se encuentre activo. Cuando se activa la rama 1 los transistores  $Q_1$  y  $Q_2$  se encuentran cerrados y el modelo queda descrito por las siguientes ecuaciones en diferencias para la corriente y la tensión respectivamente:

$$i_L(k) = i_L(k-1) + (v_g - v_{out}) \cdot \frac{dt}{L}$$

$$v_{out}(k) = v_{out}(k-1) + (i_L - i_R) \cdot \frac{dt}{C}$$

Cuando se da el caso opuesto con la rama 2 activa, los transistores  $Q_3$  y  $Q_4$  se encuentran cerrados y las ecuaciones en diferencias que describen el modelo en este caso son las siguientes:

$$i_L(k) = i_L(k-1) + (-v_g - v_{out}) \cdot \frac{dt}{L}$$

$$v_{out}(k) = v_{out}(k-1) + (i_L - i_R) \cdot \frac{dt}{C}$$

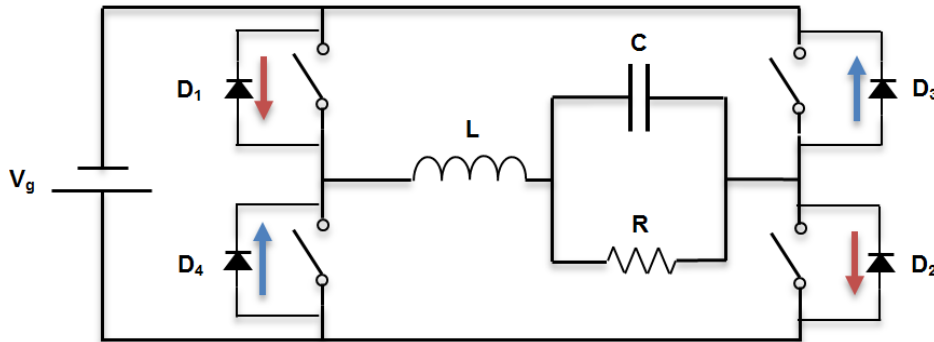


Figura 4.2. Circulación de la corriente durante los estados intermedios de conmutación

Como se ha comentado, los reguladores suelen añadir tiempos muertos para evitar cortocircuitos, por lo que el modelo debe calcular el comportamiento del inversor cuando todos los transistores se encuentran abiertos. En ese caso, la corriente circula por los diodos en antiparalelo de los MOSFET (figura 4.2). Cuando la corriente en la bobina es menor que cero ( $i_L < 0$ ) circula corriente por los diodos  $D_3$  y  $D_4$  siendo válidas las mismas ecuaciones que para la rama 1, y cuando la corriente es mayor que cero se produce el caso contrario circulando corriente por los diodos  $D_1$  y  $D_2$  siendo válidas las ecuaciones de la rama 2.

## 4.2 Implementación del modelo de la planta con señales de tipo *Real*

A partir de las ecuaciones en diferencias obtenidas en el apartado anterior, el siguiente paso es implementar dicho modelo de manera digital mediante lenguaje de descripción *hardware*, como VHDL, con el objetivo de transferir todo el diseño a un dispositivo digital programable, tal como una FPGA, y poder realizar las simulaciones y verificaciones necesarias del sistema de control. La elección de los formatos que se den a las señales digitales de dicho modelo y cómo operar con ellas no es algo trivial (tal y como se verá en el siguiente apartado), por ello, en este apartado se describe una primera aproximación al modelado de la planta utilizando señales de tipo *real*.

Este tipo de señales no son sintetizables y por tanto el diseño no puede implementarse en un dispositivo digital real, pero nos permite abstraernos de los formatos de las señales y las operaciones que se hagan con ellas y es muy rápido y sencillo de implementar, pudiendo así comprobar de forma rápida que el modelo de la planta planteado es correcto.

Además de comprobar el funcionamiento de la planta y el regulador en una primera aproximación, nos permitirá comparar los tiempos de simulación que se obtengan con este modelo con los modelos de emulación basados en HIL, tal y como se comenta en el capítulo 5 de resultados.

En las siguientes líneas se muestra el código VHDL que implementa el modelo de planta aquí descrito. El proceso *CALCULAR* realiza el cálculo del incremento de tensión y de corriente en cada instante de tiempo de las variables sin escalar (sin usar los factores  $\frac{dt}{C}$  ni  $\frac{dt}{L}$ ) en función de las señales de control de los transistores. El proceso *ACTUALIZAR* realiza el cálculo en cada ciclo de reloj de la tensión de la salida y la corriente en la bobina, aplicando los factores  $\frac{dt}{C}$  y  $\frac{dt}{L}$ .

Como se puede observar, la implementación en VHDL del modelo en coma flotante *real* es muy sencillo, necesitando únicamente transcribir el modelo en código y sin tener en cuenta la aritmética usada, ya que se considera que el tipo de datos *real* tiene resolución suficiente para todas las operaciones del modelo.



```

ACTUALIZAR: process(clk,reset)
begin
  if reset = '1' then
    VoAux <= 0.0;
    iLAux <= 0.0;

    elsif clk = '1' and clk' event then
      VoAux <= VoAux + (VoAdd*(dt/C));
      iLAux <= iLAux + (iLAdd*(dt/L));

    end if;

end process ACTUALIZAR;

```

```

CALCULAR: process(Q1,Q2,Q3,Q4,iLAux,iR,Vg,Voaux)
begin
  if Q1 = '1' and Q2 = '1' then -- rama 1
    VoAdd <= iLAux - iR;
    iLAdd <= Vg - Voaux;
  elsif Q3 = '1' and Q4 = '1' then -- rama 2
    VoAdd <= iLAux - iR;
    iLAdd <= -VoAux - Vg;
  end if;
end process CALCULAR;

```

### 4.3 Implementación del modelo de la planta con señales en coma fija

Para representar las señales que intervienen en el regulador y los coeficientes de la ecuación en diferencias se ha utilizado notación en coma fija, también conocida como notación QX.Y. Esta notación representa un número de formato fijo en complemento a dos, siendo X el número de bits que corresponden a la parte entera e Y el número de bits que corresponden a la parte decimal, añadiéndose además un bit para el signo.

Un ejemplo de representación de números en este formato sería el siguiente: por ejemplo, el número 0,6321 representado en formato Q0.7 corresponde al número binario 01010001, donde el primer bit se añade para expresar el signo sin perder rango de representación, cero bits a la parte entera y los siete restantes a la parte decimal. Se debe tener en cuenta que existe un error por pérdida de resolución, dado que no se pueden utilizar infinitos bits para representar las señales, y en el mejor de los casos sólo se puede redondear al número más cercano representable con Y cifras decimales. En el ejemplo presentado el número binario en coma fija corresponde realmente a 0,6328 y no 0,6321, por lo que se ha cometido un pequeño error, pero que es importante puesto que el modelo va acumulándolo.

Este error de resolución también existirá cuando se utilizan señales de tipo *real*, pero en este tipo de señales usan coma flotante de doble precisión (64 bits) por lo que el rango de representación es mucho mayor y el error es mucho menor que cuando se emplean señales en formato de coma fija. Como se ha comentado, es importante tener en cuenta estos errores sobre todo en sistemas de control realimentados en los que se produce la propagación y acumulación del error. Por tanto, a la hora de diseñar el modelo en coma fija es importante observar la resolución de todas las variables para que el error sea suficientemente pequeño para que el modelo sea preciso.

A la hora de representar un número en formato QX.Y mediante lenguaje VHDL se utiliza el tipo de señal vector con signo (*std\_logic\_vector*) de longitud  $X+Y+1$ . La suma de dos números en formato de coma fija definidos como QA.B y QC.D dará como resultado otra señal en coma fija de formato  $Q(\text{MAX}(A,C)+1).(\text{MAX}(B,D))$ , es decir, la longitud de la parte decimal corresponde a la del operando cuya parte decimal es de mayor longitud, y de forma similar para la parte entera pero añadiendo un bit que corresponde al bit de signo. De manera análoga, la multiplicación resultante de dos números en formato QA.B y QC.D respectivamente será un número en notación  $Q(A + C + 1).(B + D)$ . Para transformar un número en formato QX.Y a su valor decimal real hay que multiplicar dicho número por  $2^{-Y}$ .

El uso de la notación QX.Y no es interpretado por el lenguaje de descripción *hardware*, que trata a las señales como vectores convencionales en complemento a 2, y es labor del programador interpretar el valor y elegir el formato de las señales resultantes, pero con la ventaja de que es posible abstraerse de las operaciones aritméticas siempre y cuando los vectores se definan con la longitud adecuada.

Así pues, implementar un modelo utilizando señales formato QX.Y exige mucho esfuerzo de diseño comparado con el uso de señales de tipo *real* pero ofrece una velocidad de simulación y aprovechamiento del área mucho mayores.

#### 4.3.1 Escalado de señales en coma fija

Analizando las ecuaciones del apartado 4.1 observamos que al realizar su implementación en *hardware* está compuesto por operaciones de suma y multiplicación, y también de multiplexores, al existir ramas que no se activan simultáneamente. Lo más complejo de implementar en *hardware* y lo más costoso en área y por tanto en tiempo son las multiplicaciones. Si se analiza el supuesto *hardware* utilizando las ecuaciones en diferencias originales vemos que existen dos caminos críticos iguales que incluyen un multiplicador, un multiplexor y dos sumadores y, como es sabido, el camino crítico es el que limita la velocidad de emulación. Las transformaciones descritas en este apartado intentan tienen el objetivo de reducir los caminos críticos optimizando las operaciones de multiplicación (e incluso

prescindiendo de una de ellas) para reducir su impacto en área y por tanto en la velocidad de emulación.

Para optimizar el formato y tamaño de las señales empleadas y las operaciones realizadas con ellas se propone realizar una transformación o escalado. En vez de utilizar la corriente de la bobina  $i_L$  se emplea la señal  $i_L^*$  definida como:

$$i_L^* = i_L \cdot \frac{L}{dt}$$

El hecho de utilizar esta transformación nos permite expresar la ecuación en diferencias para la corriente en la bobina en la rama 1 como:

$$i_L^*(k) = i_L^*(k-1) + (v_g - v_{out})$$

Y de manera similar la corriente escalada en la bobina en la rama 2 es:

$$i_L^*(k) = i_L^*(k-1) + (-v_g - v_{out})$$

Podemos realizar una transformación análoga para la tensión en el condensador o tensión de salida, empleando la señal  $V_{out}^*$ :

$$v_{out}^* = v_{out} \cdot \frac{C}{dt}$$

Quedando la ecuación de la tensión de salida en el inversor (la misma para ambas ramas):

$$v_{out}^*(k) = v_{out}^*(k-1) + (i_L - i_R)$$

Dado que en el inversor el cálculo de la tensión de salida depende del valor de la corriente en la bobina, si se quiere usar el valor escalado de la corriente  $i_L^*$  tenemos que realizar otra transformación, empleando la señal  $v_{out}^{**}$ :

$$v_{out}^{**} = v_{out}^* \cdot \frac{C}{dt} \cdot \frac{L}{dt}$$

De esta forma la tensión de salida en el instante  $k$  se calcula como:

$$v_{out}^{**}(k) = v_{out}^{**}(k-1) + (i_L^* - i_R^*)$$

Esta transformación implica que también es necesario hacer un escalado en la corriente en la resistencia:

$$i_R^* = i_R \cdot \frac{L}{dt}$$

Dado que el modelo está realimentado, para poder cerrar el lazo de la planta es necesario eliminar el escalado para calcular el nuevo valor de tensión de salida. En la ecuación

que describe  $i_L^*(k)$  se utiliza  $v_{out}$  y no  $v_{out}^{**}$ , por lo que es necesario deshacer la transformación, lo que se hace multiplicando la señal  $v_{out}^{**}$  por el inverso de la escala, es decir:

$$v_{out} = v_{out}^{**} \cdot \frac{dt}{L} \frac{dt}{C}$$

No es necesario calcular el valor de la corriente en la bobina sin escalar para el funcionamiento del modelo, por lo que se evita tener que realizar una de las multiplicaciones. En cualquier caso, puede interesar obtenerlo (multiplicando por  $\frac{dt}{L}$ ) para monitorizar el funcionamiento de la misma durante la emulación y verificación del controlador, pero no afectará al camino crítico ni a la velocidad de emulación, puesto que no es parte de la realimentación del modelo.

Con este método de escalado se consigue prescindir de una de las multiplicaciones y por tanto uno de los caminos críticos del diseño, lo cual implica que a la hora de sintetizar el modelo, las herramientas de *place & route* podrán optimizar mejor el diseño y la emulación será más rápida.

La tabla 4.2 muestra el formato de las señales que se han empleado, indicando además para cada una de ellas el rango, la resolución y la escala aplicada. En paréntesis en la columna de señal, se muestra el nombre que se ha asignado a la señal *std\_logic\_vector* correspondiente en el código VHDL.

Como se puede ver, las señales que almacenan las variables de estado exigen tener gran resolución ya que deben almacenar tanto los valores instantáneos de la tensión y corriente (decenas de voltios o amperios) como los pequeños incrementos (microvoltios o microamperios) que integran a través de la bobina y el condensador. Como se ve, hay varios órdenes de magnitud entre los valores instantáneos y sus incrementos, por lo que hace falta usar señales de gran ancho de bits para guardar la información necesaria. Sin embargo, las señales que sirven para la realimentación sólo necesitan representar los valores instantáneos. Eso permite que el número de bits para las señales realimentadas sea menor y que el modelo pueda tener mayor velocidad de emulación.

Como se ha comentado, la tensión que se realimenta o la extracción de la corriente se puede realizar con menor número de bits. De hecho, como requieren multiplicaciones, se han limitado los anchos esas señales a 18 bits para forzar al sintetizador a utilizar bloques multiplicadores de 18 bits y poder obtener mejor velocidad y aprovechamiento de área.

Una vez obtenidas las ecuaciones en diferencias y fijados los formatos de cada señal se puede implementar el modelo de manera digital mediante lenguaje VHDL. Las siguientes líneas muestran el código de la planta. El proceso *REGISTROS* realiza el registro y actualización de los valores de tensión y corriente en cada ciclo de reloj. El proceso asíncrono *MUX* realiza la multiplexación para seleccionar las señales adecuadas en función de las señales de control (rama en la que esté funcionando el inversor). Por último, el resto del código muestra las operaciones que se realizan de manera combinacional para la implementación del modelo. El

valor de la constante de escalado para la tensión de salida es calculado previamente y se añade directamente en el código  $\frac{dt}{L} \frac{dt}{C}$  mediante una constante, siendo su formato Q29.40 y su valor literal es “010011000110”.

Señal	nº de bits	Formato	Escala	Rango	Resolución
$V_g$	13	Q6.6	-	63,984 V	0,015625 V
$V_0$ (Voaux)	13	Q6.6	-	63,984 V	0,015625 V
$i_L^*$ (iLast)	27	Q20.6	$\frac{dt}{L}$	11,651 A	$1,736 \cdot 10^{-7}$ A
$i_L^{*sat}$ (iLast1aux)	13	Q19.-7	$\frac{dt}{L}$	5,824 A	$1,423 \cdot 10^{-3}$ A
$i_R^*$ (iRast)	13	Q19.-7	$\frac{dt}{L}$	5,824 A	$1,423 \cdot 10^{-3}$ A
$V_{out}^{**}$ (Voast)	31	Q37.-7	$\frac{dt}{L} \cdot \frac{dt}{C}$	152,709 V	$1,423 \cdot 10^{-7}$ V
$V_{out}^{**sat}$ (Voast1aux)	18	Q36.-19	$\frac{dt}{L} \cdot \frac{dt}{C}$	76,354 V	$5,825 \cdot 10^{-4}$ V

Tabla 4.2. Formato, escala, resolución y rango de las señales empleadas en el modelo de planta.

```

REGISTROS: process(Clk, Reset)

    -- Cada z**-1 equivale a un registro
    begin
        if Reset = '1' then
            iLast1 <= (others => '0');
            Voast1 <= (others => '0');

            elsif Clk = '1' and Clk'event then
                --if CE = '1' then
                    iLast1 <= iLast;
                    Voast1 <= Voast;
                --end if;
            end if;
        end process REGISTROS;

```

```

--SUMA 1
Vgmaslbit <= Vg(12) & Vg;
Vomaslbit <= Voaux(12) & Voaux;
Inc1 <= Vgmaslbit - Vomaslbit;
-----

--SUMA 2
Inc2 <= - Vgmaslbit - Vomaslbit;
-----

--MUX 1
deadtm <= Inc2 when (iLast1 > 0) else
          Inc1;
-----

--SUMA 3
Incmaslbit <= Inc(13) & Inc;
iLast1_realim <= iLast1;
iLast <= Incmaslbit + iLast1_realim;
-----

--SUMA 4
iLastlaux <= iLast1(25 downto 13); --de Q20.6 a Q19.-7
iLastlauxmaslbit <= iLastlaux(12) & iLastlaux;
iRmaslbit <= iR(12) & iR;
iLmenosir <= iLastlauxmaslbit - iRmaslbit; -- Q20.-7 = Q19.-7 -
Q19.-7
-----

--SUMA 5
iLmenosirmaslbit <= iLmenosir(13) & iLmenosir;
Voast1_realim <= Voast1;
Voast <= Voast1_realim + iLmenosirmaslbit;
Voastlaux <= Voast1(29 downto 12);
-----

--PRODUCTO
Voint <= Voastlaux * dtLdtC;

```

```

MUX:PROCESS(Q1, Q2, Q3, Q4, Inc1, Inc2, deadtm)
    VARIABLE sel : STD_LOGIC_VECTOR(3 DOWNT0 0);
BEGIN
    sel := Q1 & Q2 & Q3 & Q4;

    CASE sel IS
        WHEN "1100" => Inc <= Inc1;
        WHEN "0011" => Inc <= Inc2;
        WHEN OTHERS => Inc <= deadtm;
    END CASE;

END PROCESS MUX;

```

En la figura 4.3 se muestra un diagrama del modelo de la planta que describe el código VHDL anterior, mostrando los formatos y nombres de las señales utilizadas.

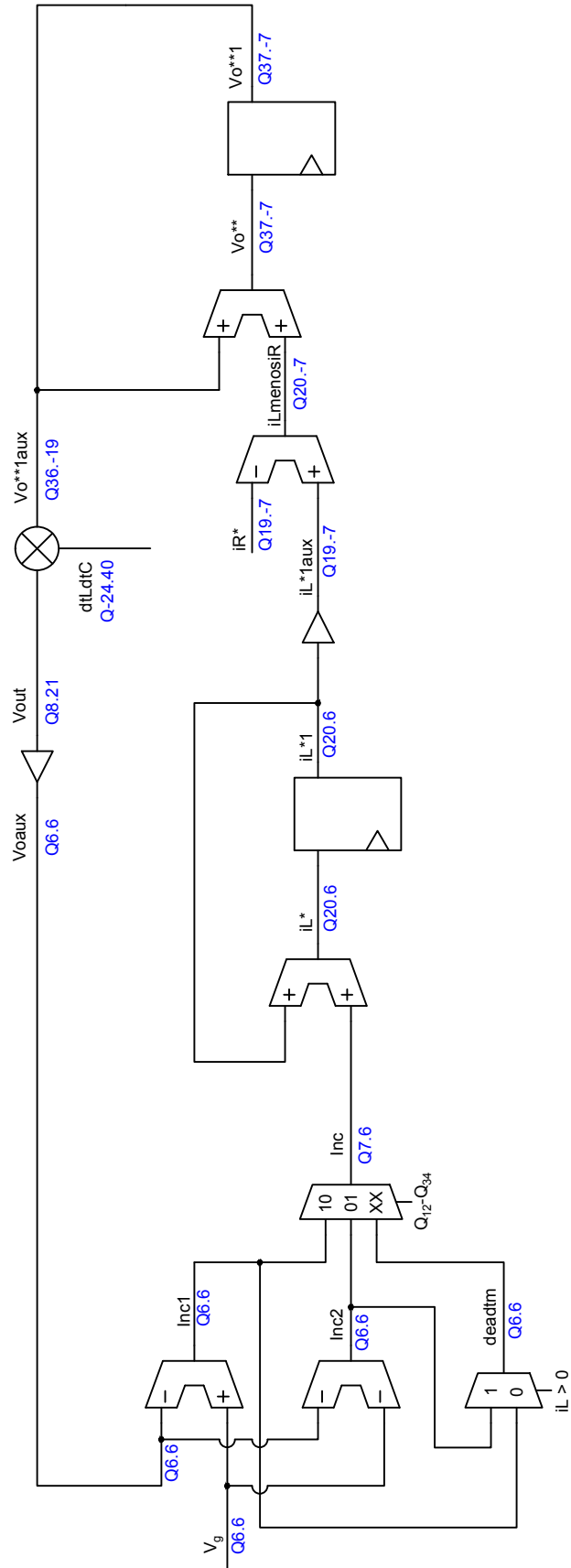


Figura 4.3. Diagrama del modelo de la planta implementado en VHDL con los formatos de las señales y escalas aplicadas

## 4.4 Implementación del regulador con señales en coma fija

De manera análoga a como se ha implementado el modelo de la planta se realiza la implementación del control del inversor, utilizando lenguaje VHDL y señales en notación QX.Y. En el apartado 3.3 se describe cuál es la función de transferencia del regulador y sus coeficientes. A partir de ella podemos obtener la siguiente ecuación en diferencias que utilizaremos para codificar el regulador (los coeficientes son los especificados en el apartado 3.3):

$$Y(k) = Y(k - 1) + A \cdot X(k) - AB \cdot X(k - 1) + AC \cdot X(k - 2)$$

En las siguientes líneas se muestra el código VHDL que implementa el regulador del inversor. Al igual que para la planta, el proceso REGISTROS registra y actualiza los valores de las señales que dependen de instantes de tiempo anteriores en cada ciclo de reloj.

```
REGISTROS: process(Clk, Reset)
-- Cada z**-1 equivale a un registro
begin
    if Reset = '1' then
        Ver <= (others => '0');
        ABVe1 <= (others => '0');
        ACVe1 <= (others => '0');
        ACVe2 <= (others => '0');
        DCaux_sat1 <= (others => '0');
    elsif Clk = '1' and Clk'event then
        if CE = '1' then
            Ver <= Ve;
            ABVe1 <= ABVer;
            ACVe1 <= ACVer;
            ACVe2 <= ACVe1;
            DCaux_sat1 <= DCaux_sat;
        end if;
    end if;
end process REGISTROS;
```

El proceso *SATURACION* reduce el ancho de la señal que contiene la información de los ciclos de trabajo de las señales de control. Además limita su valor máximo al 98% e impide que tome valores negativos. El resto de código son las operaciones aritméticas necesarias para el regulador, realizadas de manera combinacional.



```

SATURACION: process(DCaux)
begin
    if (DCaux >= DC_max) then -- satura DC_max (0.98)
        DCaux_sat <= DC_max;

    elsif (DCaux(35) = '1') then -- si es < 0, DCaux_sat vale 0
        DCaux_sat <= (others => '0');
    else
        DCaux_sat <= DCaux(35) & DCaux (33 downto 0);
    end if;
end process SATURACION;

```

```

AVer <= A * Ver;
ABVer <= AB * Ver;
ACVer <= AC * Ver;

--resta
-----
ABVelmaslbit <= ABVe1(31) & ABVe1 & "0"; -- de Q8.23 a Q9.24
ACVe2maslbit <= ACVe2(31) & ACVe2; -- de Q7.24 a Q8.24
resta <= ACVe2maslbit - ABVelmaslbit;

--primera suma
-----
restamaslbit <= resta(33) & resta; -- de Q9.24 a Q10.24
AVermaslbit <= AVer(17) & AVer & "0000000000000000"; -- de Q7.10 a Q8.24
suma <= AVermaslbit + restamaslbit;

--segunda suma
-----
sumamaslbit <= suma(34) & suma; -- de Q10.24 a Q11.24
DCAux_sat1maslbit <= DCAux_sat1(34) & DCAux_sat1; -- de Q10.24 a Q11.24
DCAux <= sumamaslbit + DCAux_sat1maslbit;

--escalado
-----
DC116 <= DCAux_sat(34) & DCAux_sat (24 downto 8); -- de Q10.24 a Q1.16
conservando signo
DC <= escala * DC116; -- Salida del regulador. Valores de PWM
escalados
-----
DC116 <= DCAux_sat(34) & DCAux_sat (24 downto 8); -- de Q10.24 a Q1.16
conservando signo
DC <= escala * DC116; -- Salida del regulador. Valores de PWM
escalados

```

En la figura 4.4 se muestra un diagrama de la función de transferencia del regulador, especificando el formato de las señales y el nombre de las mismas en el código VHDL anterior.



# 5

## Resultados

En este apartado se describirán los resultados obtenidos del sistema de verificación y emulación implementado para el control de un convertidor inversor, comparando la solución HIL que utiliza modelos de la planta y del regulador descritos en el apartado anterior utilizando señales en formato de coma fija sintetizable, con la solución que utiliza señales de tipo real y que solo permiten simulación en un ordenador. Se compararán ambos sistemas en términos de facilidad de diseño e implementación, precisión y velocidad de simulación.

### 5.1 Descripción del sistema completo.

El sistema de simulación y verificación está compuesto además de por el propio regulador y el modelo de la planta, de otros módulos que permiten el funcionamiento del inversor.

El módulo PWM recibe como entrada una señal de 32 bits que contiene la información del ciclo de trabajo que se debe aplicar a la planta y ofrece a su salida las señales de control que gobiernan los transistores de la planta. Este módulo genera además una señal de *chip\_enable* que se utiliza para actualizar el resto de módulos a la velocidad de la frecuencia de conmutación ( $f_{SW}$ ). El módulo SENO genera una señal senoidal en formato Q6.7 que se utiliza como señal de referencia para obtener la señal de error restándola con la tensión de salida realimentada. Este formato permite definir la referencia de la tensión de salida con una resolución de 0,0078125 V. Los valores de la señal seno se almacenan en las BRAM (*Block Random Access Memory*) de la FPGA y han sido calculados previamente mediante un script escrito en Matlab que permite especificar el número de pasos para cada ciclo del seno y la resolución de las señales y genera un fichero de texto con el código formateado en VHDL.

La amplitud máxima de la señal seno es 1, por lo que se multiplica por un valor de referencia fijo que fija la amplitud de la señal que ofrecerá a su salida el inversor. El módulo seno, PWM y el regulador se integran en el módulo FPGA\_TOP, puesto que son los elementos que componen el sistema de control y que se conectarán a un inversor real una vez simulado y verificado el sistema completo.

```
Nvalores=400; % número de pasos para el seno
Nhex=5; %numero de cifras hexadecimales
Nbits=16; % bits de resolución
VHDLvar='contador_seno';%variable de contador en fichero VHDL

t=[0:((2*pi)/Nvalores):2*pi];
seno=sin(t);
figure
plot(t,seno);

%Cuantificación de valores del seno
%-----
min=min(seno);
rango=abs(min-1);
res=1/(2^Nbits);
vector=0:res:1; %escalones para valores del seno
for i=1:Nvalores
    for k=1:length(vector)
        if abs(seno(i))>vector(k)
            cuant(i)=uint32((k-1));
        end
    end
end
figure;
u=[1:Nvalores];
plot(u,cuant,'o');
title('\bfCiclo cuantificado con Nbits');

%Conversión de valores del seno a Q1.16 hexadecimal
%-----
for i=1:Nvalores
    if seno(i)>= 0 %valores negativos del seno
        comafija(i,:)= dec2hex(cuant(i),Nhex);
    else
        comafija(i,:)= dec2hex(bitcmp(abs(cuant(i))-1,Nbits+2),Nhex);
    end
end
comafija2 = cellstr(comafija);
```

El modelo de la planta se conecta a FPGA\_TOP recibiendo como entradas las cuatro señales que controlan los MOSFET, la señal  $V_g$ , que es la tensión continua de entrada que recibe el inversor y la señal  $i_R$ , que es la corriente en la carga. Considerando una carga resistiva se puede obtener como el producto de la tensión de salida por el valor elegido para la carga. Ha de tenerse en cuenta que este valor de corriente está escalado tal y como se indica en el **apartado 4.3.1**. Las salidas son la tensión de salida  $V_o$  y la corriente en la bobina  $i_L$ , ambas en



$R_{\text{carga}}$	$L$	$C$	$V_g$	$f_{\text{CLK}}$	$f_{\text{SW}}$	$f_{\text{CICLO}}$
12 $\Omega$	0,9 mH	0,1 mF	20 V	100 MHz	20 kHz	50 Hz

Tabla 5.1. Parámetros de diseño escogidos del entorno de simulación HIL.

Las formas de onda que se presentan a continuación corresponden a unos parámetros de diseño específicos que se indican en la tabla 5.1. La figura 5.2 muestra las formas de onda de la corriente de salida del inversor, la tensión de salida y la tensión en la bobina usando señales de tipo *real*, tanto para el modelo de la planta y como para el regulador, durante dos periodos de red (40 ms) aplicando una tensión de referencia de 15 V. Se puede apreciar como la tensión de salida es una señal sinusoidal de amplitud máxima cercana a 15 y un transitorio durante el tiempo inicial además de cierto rizado en la corriente. La figura 5.3 muestra resultados similares usando en este caso el regulador sintetizable con señales en notación QX.Y y la planta modelada con señales *real*.

La figura 5.4 muestra capturas de las mismas señales de onda utilizando los mismos parámetros, pero con el modelo sintetizable tanto para regulador como para la planta. Para poder visualizar las señales, los valores que se representan son las propias señales QX.Y convertidas a formato real mediante la instrucción *conv\_integer* de VHDL tal y como muestra la siguiente línea de código:

```
señal_real <= real (conv_integer(señal_QXY))*escala*(2.0**Y);
```

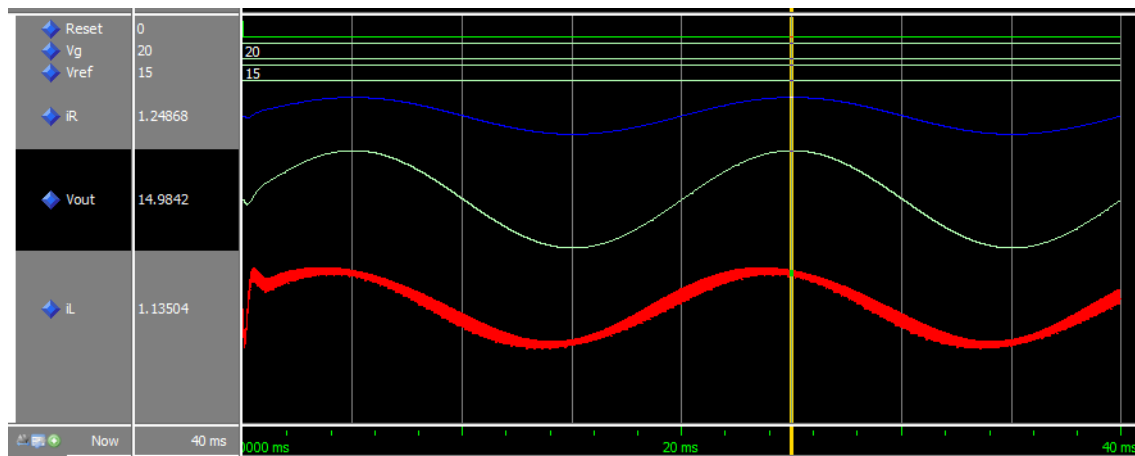


Figura 5.2. Formas de onda para la tensión de salida, corriente en la bobina y corriente en la resistencia usando señales de tipo *real*

Para comparar de forma objetiva los resultados obtenidos se han tomado valores de las variables de entorno, es decir, tensión y corriente en ciertos instantes de tiempo durante un ciclo de red. Estos datos se han obtenido con ambos modelos de la planta y se ha calculado el error absoluto entre dichos valores.

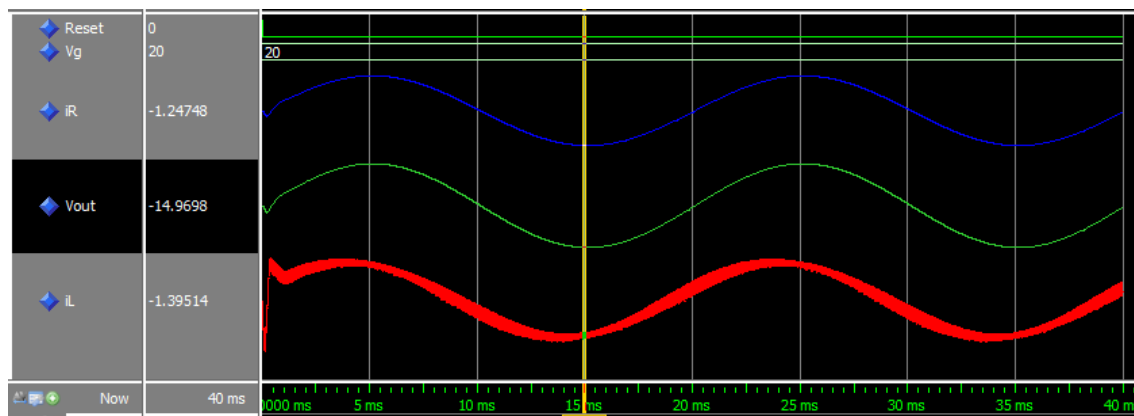


Figura 5.3. Formas de onda para la tensión de salida, corriente en la bobina y corriente en la resistencia usando un regulador sintetizable y señales de tipo *real* para el modelo de la planta.

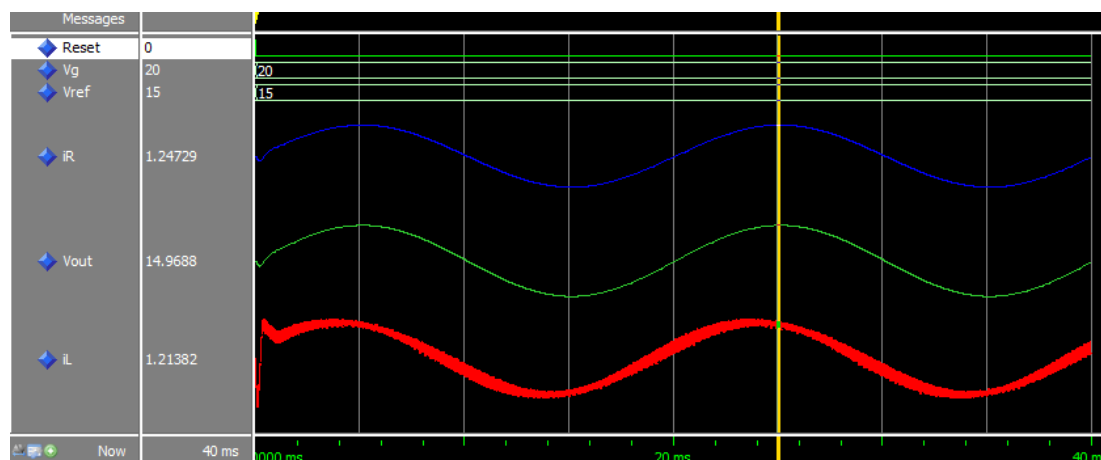


Figura 5.4. Formas de onda para la tensión de salida, corriente en la bobina y en la resistencia usando señales en formato QX.Y.

Primeramente analizamos el comportamiento del sistema teniendo en cuenta únicamente el modelo de la planta en lazo abierto. La simulación en lazo abierto prescinde del regulador y se utiliza un ciclo de trabajo fijo. Al considerar solo la planta sin el regulador y los demás elementos podemos verificar realmente que la planta se ha modelado adecuadamente respecto al modelo *real*, pues con el sistema en lazo cerrado pueden enmascarse errores. Por ejemplo un mal modelado de la planta que provoque un valor de tensión mayor del esperado puede no ser apreciado ya que el regulador actuaría proporcionando la tensión adecuada.

En la tabla 5.2 se muestra la comparativa entre los resultados obtenidos con las simulaciones de *ModelSim* con señales de tipo *real* y el modelo sintetizable que emplea señales en formato QX.Y para tensión de salida con la planta en lazo abierto. El valor de ciclo de trabajo empleado en este caso es 0,5. Como se observa la precisión del modelo es elevada siendo el error medio absoluto en la tensión de 0,02313 V. La tabla 5.3 muestra la misma comparativa en este caso para la corriente en la bobina. El error medio en la corriente es de 0,00053.

t (ms)	0,5	1	2	3	4	5	6	7	8
<b>v<sub>out</sub> (V)</b>									
ModelSim Real (lazo abierto)	0,6803	-0,0910	0,1189	-0,1145	0,0978	-0,0769	0,0580	-0,0415	0,0293
<b>v<sub>out</sub> (V)</b>									
ModelSim QX.Y (lazo abierto)	0,6875	0,0781	0,1094	-0,1094	0,0938	-0,0781	0,0625	-0,0469	0,0313
<b>Error Abs.</b> (lazo abierto)	0,0072	0,1691	0,0095	0,0052	0,0041	0,0012	0,0045	0,0054	0,0020

Tabla 5.2. Valores de tensión de salida para el modelo en lazo abierto en los sistemas con señales de tipo real y con señales en formato QX.Y

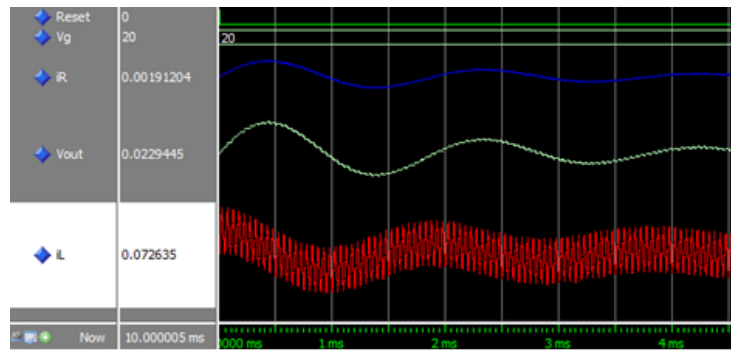
t (ms)	0,5	1	2	3	4	5	6	7	8
<b>i<sub>L</sub> (A)</b>									
ModelSim Real (lazo abierto)	-0,26577	-0,46004	-0,15630	-0,35030	-0,23002	-0,30213	-0,26063	-0,28325	-0,27187
<b>i<sub>L</sub> (A)</b>									
ModelSim QX.Y (lazo abierto)	-0,26358	-0,46027	-0,15579	-0,35075	-0,22961	-0,30225	-0,26030	-0,28356	-0,27161
<b>Error Abs.</b> (lazo abierto)	0,00219	0,00023	0,00051	0,00045	0,00041	0,00013	0,00033	0,00030	0,00025

Tabla 5.3. Valores de corriente en la bobina para el modelo en lazo abierto en los sistemas con señales de tipo real y con señales en formato QX.Y

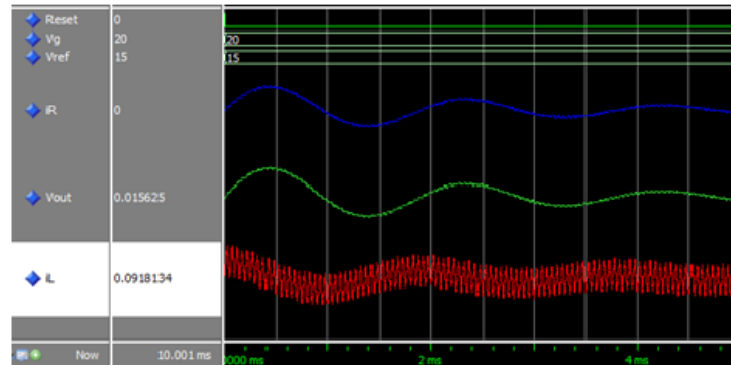
En la figura 5.5 se muestran las formas de onda para la corriente en la resistencia, tensión de salida y corriente en la bobina para el modelo de planta en lazo abierto con un ciclo de trabajo de 0,5. La figura 5.5.a corresponde al modelo que usa señales *real* mientras que la figura 5.5.b muestra el caso análogo con señales en formato QX.Y.

Por otra parte las figuras 5.6 y 5.7 muestran una comparativa similar a la anterior usando valores de ciclo de trabajo 0,1 y 0,9 respectivamente



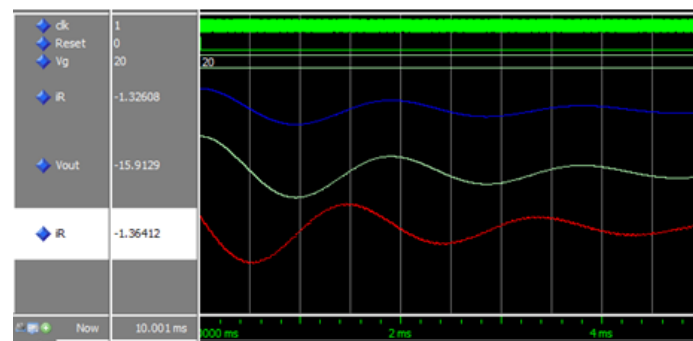


(a)

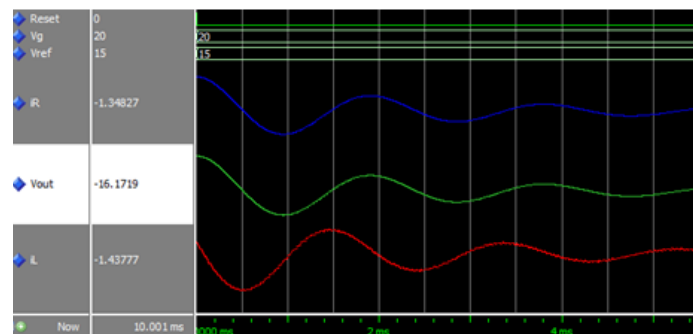


(b)

Figura 5.5. Formas de onda del modelo en lazo abierto con ciclo de trabajo 0,5 usando señales *real* (a) y en formato QX.Y (b)

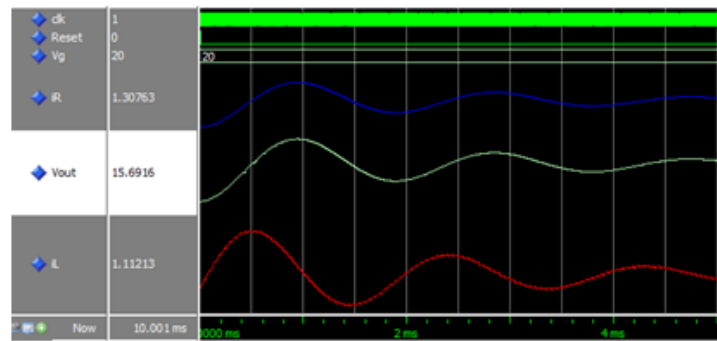


(a)

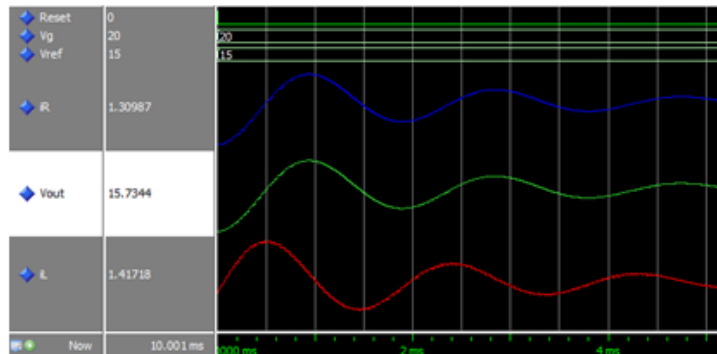


(b)

Figura 5.6. Formas de onda del modelo en lazo abierto con ciclo de trabajo 0,1 usando señales *real* (a) y en formato QX.Y (b)



(a)



(b)

Figura 5.7. Formas de onda del modelo en lazo abierto con ciclo de trabajo 0,9 usando señales *real* (a) y en formato QX.Y (b)

Una vez comprobada la precisión del modelo en lazo abierto realizamos pruebas similares a las anteriores pero considerando todo el sistema completo, es decir, la planta más el sistema de control, el módulo PWM, el generador de señales sinusoidales y demás elementos. Esta prueba es la más realista, puesto que prueba el sistema completo HIL, junto al regulador a depurar.

La tabla 5.4 muestra la comparativa entre los resultados obtenidos con las simulaciones de *ModelSim* con señales de tipo *real* y el modelo sintetizable que emplea señales en formato QX.Y para tensión de salida para el sistema completo en lazo cerrado. El error absoluto medio en los valores tomados es de 0,292 V, lo que corresponde a un 1,94% de error sobre el valor máximo de la tensión de salida. La tabla 5.5 muestra la misma comparativa para los valores de corriente en la bobina, siendo el error medio absoluto de 0,052 A, lo que representa aproximadamente un 0,23% de error respecto al valor máximo de corriente.

La figura 5.8 muestra cómo se comportan las señales de tensión y corriente para distintos valores de la tensión de referencia, en concreto 15 V, 20 V, 5 V y 10 V al inicio de cuatro ciclos de red consecutivos. Como se puede ver en las formas de onda la tensión de salida sigue adecuadamente el valor de tensión de referencia aplicada.

En la figura 5.9 se observa cómo responde el sistema ante una tensión de referencia excesiva, produciéndose la saturación de la corriente y tensión de salida.

t (ms)	0,5	2,5	5	7,5	10	12,5	15	17,5	20
$v_{out}$ (V) ModelSim Real	0,621	10,417	14,984	10,772	0,244	-10,416	-14,981	-10,771	-0,246
$v_{out}$ (V) ModelSim QX.Y	0,234	10,078	14,669	11,094	0,119	-10,078	-14,969	-11,094	-0,719
Error Abs.	0,386	0,339	0,315	0,322	0,125	0,338	0,012	0,323	0,473

Tabla 5.4. Valores de tensión de salida para el modelo en lazo cerrado en los sistemas con señales de tipo real y con señales en formato QX.Y.

t (ms)	0,5	2,5	5	7,5	10	12,5	15	17,5	20
$i_L$ (A) ModelSim Real	0,917	1,009	1,356	0,371	-0,726	-1,405	-1,380	-0,771	0,174
$i_L$ (A) ModelSim QX.Y	0,905	0,979	1,124	0,413	-0,689	-1,391	-1,396	-0,811	0,125
Error Abs.	0,012	0,030	0,232	0,042	0,037	0,014	0,016	0,039	0,050

Tabla 5.5. Valores de tensión de salida para el modelo en lazo cerrado en los sistemas con señales de tipo real y con señales en formato QX.Y.

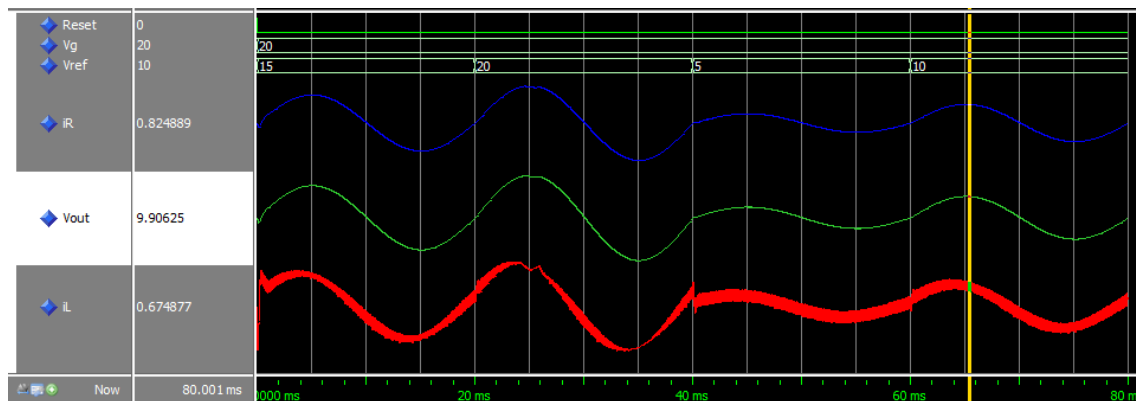


Figura 5.8. Tensión y corriente de salida en el inversor frente a distintos valores de tensión de referencia

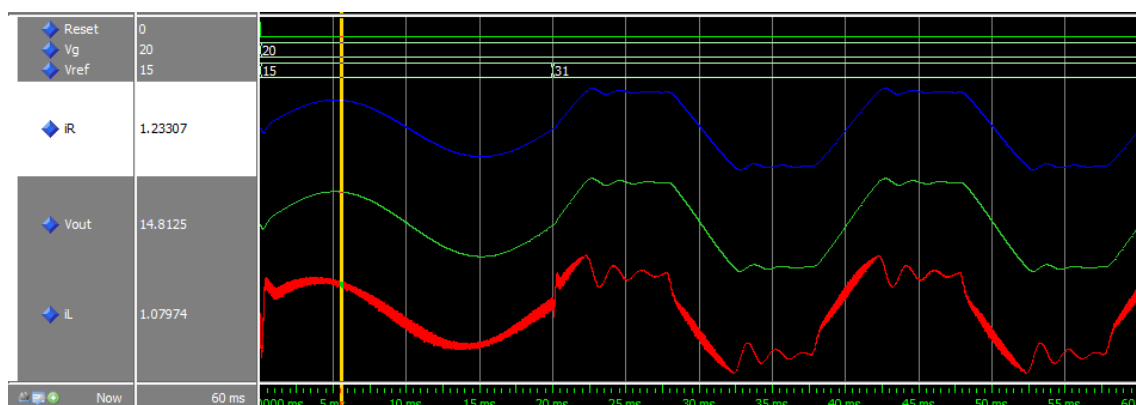


Figura 5.9. Saturación del inversor al aplicar un valor excesivo de tensión de referencia

### 5.3 Velocidad de simulación y emulación.

Uno de los motivos principales por los que emplear un entorno HIL es la alta velocidad de emulación que se puede alcanzar en comparación con otros métodos de simulación. Demostrada la precisión del sistema planteado, en este apartado se realiza una comparación entre las velocidades alcanzadas en simulación y emulación.

El tiempo que tarda en llevarse a cabo una simulación con *ModelSim* del sistema de control implementando la planta mediante señales *real* durante un ciclo de red (20 ms) es de aproximadamente 24 s. La simulación se ha llevado a cabo en un ordenador con procesador *Intel i5* a 2,4 GHz y 4 GB de RAM bajo *Windows 7 Service Pack 1* de 64 bits. La simulación con *ModelSim* utilizando el modelo de planta en coma fija es de aproximadamente 12 s. Es decir, la simulación de un modelo en coma fija presenta una aceleración de 2x frente a una simulación en coma flotante no sintetizable. Sin embargo, el modelo en coma fija puede emularse, permitiendo aceleraciones mucho mayores.

El tiempo teórico que se podría alcanzar con la emulación, es decir, con la planta y el regulador implementados en coma fija en una FPGA, viene determinado por el mayor retardo en el camino crítico una vez realizado el *place and route*. En este caso dicho tiempo es de 13,959 ns (71,638 MHz). Este tiempo se ha obtenido realizando implementando el diseño sobre una FPGA Virtex 4 de Xilinx. Dado que el tiempo de integración del modelo del inversor es 10 ns, y que se quieren simular 20 ms, el sistema HIL debe calcular 2000000 pasos de integración. El sistema emulado tardaría 20 ms si el tiempo crítico fuera igual al tiempo de integración, pero como el tiempo crítico es 13,959 ns, una simulación de 20 ms tardaría 27,918 ms. Como vemos el tiempo de emulación es varios órdenes de magnitud menor que el de simulación, obteniendo una aceleración de 859x aproximadamente. Aquí es donde radica la ventaja del uso de un sistema HIL emulado.

En la tabla 5.6 se muestra la utilización de los recursos en el dispositivo al implementar el sistema completo (LUTs, flip-flops y multiplicadores utilizados) y el modelo de la planta solamente. Como se puede comprobar el modelo de la planta aislado requiere un área mucho menor que el sistema completo. Estos valores indican que el este sistema de verificación HIL necesita de pocos recursos y se puede implementar fácilmente en una FPGA de bajo coste, lo que lo hace ideal para el prototipado y depuración.

	Flip-flops	LUTs	MULT 18x18
Sistema completo	244 (2%)	997 (9%)	5 (15%)
Modelo de la planta	57 (1%)	122 (1%)	1 (3%)

Tabla 5.6. Utilización de recursos de la FPGA para el modelo de planta y para el sistema HIL completo

## 5.4 Comparativa

En los apartados anteriores se ha comprobado que el sistema de emulación desarrollado en este trabajo fin de Máster funciona con la precisión adecuada en comparación al sistema que utiliza señales de tipo real y nos permite verificar el sistema de control en tiempos varios ordenes de magnitud menores, casi permitiendo emular en tiempo real.

Podemos concluir que el método de emulación HIL utilizando señales en coma fija emulado es el que permite mayor velocidad, del orden de milisegundos, pero es el que exige un mayor esfuerzo por parte del diseñador, en comparación con el método de simulación mediante señales de tipo *real*, que ofrece tiempos del orden de segundos pero el esfuerzo de programación es considerablemente menor, sin tener que preocuparse de anchos y formatos de cada una de las señales, operaciones aritméticas y factores de escala. En cualquier caso, es importante recordar que el diseño del modelo se realizará únicamente una vez, mientras que la depuración del presente y futuros reguladores se puede realizar múltiples veces, y precisamente esa etapa es la que mejora enormemente la emulación HIL.



# 6

## Conclusiones y trabajo futuro

### 6.1 Conclusiones

En el proceso de diseño de un controlador digital para un convertidor de potencia es imprescindible un método para verificar su correcto funcionamiento antes de su puesta en marcha debido al riesgo inherente al trabajar con altas tensiones y corrientes. En este trabajo se ha demostrado la validez de la técnica de emulación HIL para la depuración de un convertidor inversor DC/AC con topología full-bridge realizando un modelo digital del mismo mediante lenguaje de descripción *hardware* VHDL. Así mismo se ha diseñado el regulador y todos los elementos adicionales necesarios como el módulo de pulsos PWM y el generador de señal sinusoidal de referencia, para comprobar la validez del sistema. Para ello ha sido necesario obtener previamente modelos matemáticos para la planta y el regulador.

Para realizar un modelo emulable de la planta se han utilizado señales en coma fija, específicamente en notación QX.Y, y para optimizar el diseño se han introducido optimizaciones *hardware*, descritas en detalle en esta memoria, que permiten prescindir de ciertas operaciones aritméticas y aprovechar mejor los recursos de la FPGA en cuanto a velocidad y área.

A lo largo de diferentes pruebas, se han comparado los resultados obtenidos del sistema HIL con un sistema de simulación que utiliza señales en coma flotante de tipo *real*. El tipo real implementa el estándar IEEE 754 en su versión de doble precisión, utilizando 52 bits de mantisa, por lo que obtiene una gran resolución. Los resultados muestran que el modelo sintetizable comete errores absolutos muy pequeños, demostrando la validez de la metodología HIL. Además se ha comprobado que esta técnica permite unas velocidades de emulación mucho mayores utilizando pocos recursos digitales cuando se diseñan óptimamente. En contrapartida, el esfuerzo requerido en el diseño comparando con el sistema que usa señales de tipo *real* es considerablemente mayor, pues requiere por parte del programador ajustar cada una de los formatos y anchos de las señales en el modelo de la planta. En cualquier caso, el esfuerzo de diseño sólo debe realizarse una vez al diseñar el

modelo, mientras que el aumento de la velocidad de las pruebas se aprovecha durante todo la etapa de pruebas del presente regulador o de futuros reguladores.

En resumen, se ha demostrado la validez y utilidad de la técnica de verificación HIL para depurar el funcionamiento de un convertidor inversor gobernado por un circuito de control digital, todo ello modelado mediante lenguaje de descripción *hardware* para ser implementado en una FPGA. Se ha comprobado que la simulación de un modelo en coma fija presenta una aceleración de 2x frente a una simulación en coma flotante no sintetizable. Sin embargo, el modelo en coma fija puede emularse, permitiendo aceleraciones mucho mayores.

## 6.2 Trabajo futuro

De los resultados obtenidos en este trabajo fin de Máster surgen nuevas líneas de investigación y mejora relacionada con la verificación HIL de sistemas de control digitales. Las líneas de investigación futura y posibles mejoras propuestas sobre el trabajo realizado son las siguientes:

- Comparar los métodos propuestos de simulación con variable *real* y emulación con señales QX.Y con otros métodos, como el que utiliza señales en formato *sfixed* o la utilización de herramientas de simulación mixta analógico-digital, para comprobar las características de otros sistemas de verificación de sistemas de control digitales.
- Diseñar y realizar un prototipo del circuito de un convertidor inversor para comparar el funcionamiento del sistema de emulación descrito en este trabajo con un sistema físico real.
- Realizar un estudio paramétrico más exhaustivo del funcionamiento del sistema de emulación frente a cambios en distintas variables, como variaciones en la carga, en la tensión de entrada, resolución de la señal senoidal de referencia, frecuencia de conmutación, etc.
- Implementar los modelos de la planta, tanto para simulación como emulación, teniendo en cuenta pérdidas, según las ecuaciones obtenidas en el apartado 3.2
- Extender el estudio realizado a otras aplicaciones que utilicen técnicas HIL para la verificación de la regulación de circuitos electrónicos de potencia, como el control de otro tipo de convertidores, distintas topologías de inversores, control de motores, etc.
- Realizar un estudio similar para un convertidor inversor trifásico y verificar sobre el la validez el sistema de emulación HIL.



## Glosario de Abreviaturas

AC	Alternating Current
ADC	Analog-to-Digital Converter
ASIC	Application Specific Integrated Circuit
BRAM	Block RAM
DC	Direct Current
DSP	Digital Signal Processor
FPGA	Field-Programmable Gate Array
HDL	Hardware Description Language
HIL	Hardware In-the-Loop
LUT	Look-Up Table
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
PWM	Pulse-Width Modulation
RAM	Read Only Memory
VHDL	VHSIC Hardware Description Language
VHDL-AMS	VHDL Analog and Mixed-Signal
VHSIC	Very-High-Speed Integrated Circuits



## Bibliografía

- [1] A. De Castro. Aplicación del Control Digital basado en Hardware Específico para Convertidores de Potencia Conmutados. *Tesis Doctoral*. 2003
- [2] A. Sanchez, A. de Castro, and J. Garrido, "A Comparison of Simulation and Hardware-in-the- Loop Alternatives for Digital Control of Power Converters," IEEE Transactions on Industrial Infomatics, vol. 8, no. 3, pp. 491–500, Aug. 2012.
- [3] A. Sánchez. Aportaciones Mediante Implementación basada en sistemas embebidos al control digital de convertidores conmutados. *Tesis Doctoral*. 2013.
- [4] O. Goni, A. Sanchez, E. Todorovich & A. de Castro, "Resolution Analysis of Switching Converter Modelsfor Hardware-in-the-Loop", in IEEE Transactions on Industrial Informatics, vol. 10, no. 2, pp. 1162-1170, may 2014.
- [5] Matlab, "www.mathworks.com," 2013.
- [6] I. Urriza, L. A. Barragán, J. I. Artigas, J. Acero, D. Navarro, and J. M. Burdío, "Using Mixed-Signal Simulation to Design a Digital Power Measurement System for Induction Heating Home Appliances," in Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on, 2007, pp. 1447–1451.
- [7] Questa Advanced Simulator, "www.mentor.com," 2011.
- [8] Matlab, "www.mathworks.com," 2013.
- [9] P. Zumel, M. García-Valderas, A. Lázaro, C. Loópez-Ongil, and A. Barrado, "Co-simulation psim-modelsim oriented to digitally controlled switching power converters," in Control and Modeling for Power Electronics (COMPEL), 2010 IEEE 12th Workshop on, pp. 1\_7, jun. 2010.
- [10] B. Lu, X. Wu, H. Figueroa, and A. Monti, "A low-cost real-time hardware- in-the-loop testing approach of power electronics controls," IEEE Trans. Ind. Electron., vol. 54, no. 2, pp. 919–931, Apr. 2007.
- [11] S. Karimi, P. Poure, and S. Saadate, "An hil-based reconfigurable platform for design, implementation, and verification of electrical system digital controllers," IEEE Trans. Ind. Electron., vol. 57, no. 4, pp. 1226–1236, Apr. 201
- [12] A. Myaing and V. Dinavahi, Fpga-based real-time emulation of power electronic systems with detailed representation of device characteristics,Industrial Electronics, IEEE Transactions on, vol. 58, pp. 358368, jan. 2011.

- [13] Ó. Lucía, I. Urriza, L. A. Barragán, D. Navarro, O. Jiménez, and J. M. Burdío, “*Real-Time FPGA-Based Hardware-in-the-Loop Simulation Test Bench Applied to Multiple-Output Power Converters*,” *IEEE Transactions on Industry Applications*, vol. 47, no. 2, pp. 853–860, 2011.
- [14] O. Lucia, O. Jiménez, L. Barragán, I. Urriza, J. Burdío, and D. Navarro, "Realtime fpga-based hardware-in-the-loop development test-bench for multiple output power converters," in *Applied Power Electronics Conference and Exposition (APEC)*, 2010 *Twenty-Fifth Annual IEEE*, pp. 309-314, feb. 2010.
- [15] G. G. Parma and V. Dinavahi, “Real-Time Digital Hardware Simulation of Power Electronics and Drives,” in *Power Engineering Society General Meeting, 2007. IEEE*, 2007, vol. 22, no. 2, pp. 1235–1246.
- [16] H. Li, M. Steurer, K. L. Shi, S. Woodruff, and D. Zhang, “Development of a Unified Design , Test , and Research Platform for Wind Energy Systems Based on,” *Industrial Electronics, IEEE Transactions on*, vol. 53, no. 4, pp. 1144–1151, 2006.
- [17] O. Goni, A. Sanchez, E. Todorovich & A. de Castro, "*Resolution Analysis of Switching Converter Modelsfor Hardware-in-the-Loop*", in *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1162-1170, may 2014.
- [18] Dragan Marksimovic R.W. Erikson. *Fundamentals of Power Electronics*. Ed. Chapman & Hall, 2 edition, 2001.